To all our customers

Regarding the change of names mentioned in the document, such as Hitachi Electric and Hitachi XX, to Renesas Technology Corp.

The semiconductor operations of Mitsubishi Electric and Hitachi were transferred to Renesas Technology Corporation on April 1st 2003. These operations include microcomputer, logic, analog and discrete devices, and memory chips other than DRAMs (flash memory, SRAMs etc.) Accordingly, although Hitachi, Hitachi, Ltd., Hitachi Semiconductors, and other Hitachi brand names are mentioned in the document, these names have in fact all been changed to Renesas Technology Corp. Thank you for your understanding. Except for our corporate trademark, logo and corporate statement, no changes whatsoever have been made to the contents of the document, and these changes do not constitute any alteration to the contents of the document itself.

Renesas Technology Home Page: http://www.renesas.com

Renesas Technology Corp.
Customer Support Dept.
April 1, 2003

Cautions

Keep safety first in your circuit designs!

1. Renesas Technology Corporation puts the maximum effort into making semiconductor products better and more reliable, but there is always the possibility that trouble may occur with them. Trouble with semiconductors may lead to personal injury, fire or property damage.
   Remember to give due consideration to safety when making your circuit designs, with appropriate measures such as (i) placement of substitutive, auxiliary circuits, (ii) use of nonflammable material or (iii) prevention against any malfunction or mishap.

Notes regarding these materials

1. These materials are intended as a reference to assist our customers in the selection of the Renesas Technology Corporation product best suited to the customer's application; they do not convey any license under any intellectual property rights, or any other rights, belonging to Renesas Technology Corporation or a third party.
2. Renesas Technology Corporation assumes no responsibility for any damage, or infringement of any third-party's rights, originating in the use of any product data, diagrams, charts, programs, algorithms, or circuit application examples contained in these materials.
3. All information contained in these materials, including product data, diagrams, charts, programs and algorithms represents information on products at the time of publication of these materials, and are subject to change by Renesas Technology Corporation without notice due to product improvements or other reasons. It is therefore recommended that customers contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor for the latest product information before purchasing a product listed herein.
   The information described here may contain technical inaccuracies or typographical errors.
   Renesas Technology Corporation assumes no responsibility for any damage, liability, or other loss rising from these inaccuracies or errors.
   Please also pay attention to information published by Renesas Technology Corporation by various means, including the Renesas Technology Corporation Semiconductor home page (http://www.renesas.com).
4. When using any or all of the information contained in these materials, including product data, diagrams, charts, programs, and algorithms, please be sure to evaluate all information as a total system before making a final decision on the applicability of the information and products. Renesas Technology Corporation assumes no responsibility for any damage, liability or other loss resulting from the information contained herein.
5. Renesas Technology Corporation semiconductors are not designed or manufactured for use in a device or system that is used under circumstances in which human life is potentially at stake. Please contact Renesas Technology Corporation or an authorized Renesas Technology Corporation product distributor when considering the use of a product contained herein for any specific purposes, such as apparatus or systems for transportation, vehicular, medical, aerospace, nuclear, or undersea repeater use.
6. The prior written approval of Renesas Technology Corporation is necessary to reprint or reproduce in whole or in part these materials.
7. If these products or technologies are subject to the Japanese export control restrictions, they must be exported under a license from the Japanese government and cannot be imported into a country other than the approved destination.
   Any diversion or reexport contrary to the export control laws and regulations of Japan and/or the country of destination is prohibited.
8. Please contact Renesas Technology Corporation for further details on these materials or the products contained therein.

H8/3048F-ONE E10T Emulator

User's Manual

ADE-702-287

Rev. 1.0
05/28/02
Hitachi, Ltd.

Cautions

1. Hitachi neither warrants nor grants licenses of any rights of Hitachi's or any third party's patent, copyright, trademark, or other intellectual property rights for information contained in this document. Hitachi bears no responsibility for problems that may arise with third party's rights, including intellectual property rights, in connection with use of the information contained in this document.
2. Products and product specifications may be subject to change without notice. Confirm that you have received the latest product standards or specifications before final design, purchase or use.
3. Hitachi makes every attempt to ensure that its products are of high quality and reliability. However, contact Hitachi's sales office before using the product in an application that demands especially high quality and reliability or where its failure or malfunction may directly threaten human life or cause risk of bodily injury, such as aerospace, aeronautics, nuclear power, combustion control, transportation, traffic, safety equipment or medical equipment for life support.
4. Design your application so that the product is used within the ranges guaranteed by Hitachi particularly for maximum rating, operating supply voltage range, heat radiation characteristics, installation conditions and other characteristics. Hitachi bears no responsibility for failure or damage when used beyond the guaranteed ranges. Even within the guaranteed ranges, consider normally foreseeable failure rates or failure modes in semiconductor devices and employ systemic measures such as fail-safes, so that the equipment incorporating Hitachi product does not cause bodily injury, fire or other consequential damage due to operation of the Hitachi product.
5. This product is not designed to be radiation resistant.
6. No one is permitted to reproduce or duplicate, in any form, the whole or part of this document without written approval from Hitachi.
7. Contact Hitachi's sales office for any questions regarding this document or Hitachi semiconductor products.

IMPORTANT INFORMATION

READ FIRST

• READ this user's manual before using this emulator product.

• KEEP the user's manual handy for future reference.

Do not attempt to use the emulator product until you fully understand its mechanism.

Emulator Product:

Throughout this document, the term "emulator product" shall be defined as the following products produced only by Hitachi, Ltd. excluding all subsidiary products.

- Emulator
- User system interface cable

The user system or a host computer is not included in this definition.

Purpose of the Emulator Product:

This emulator product is a software and hardware development tool for systems employing the Hitachi microcomputer. This emulator product must only be used for the above purpose.

Limited Applications:

This emulator product is not authorized for use in MEDICAL, atomic energy, aeronautical or space technology applications without consent of the appropriate officer of a Hitachi sales company. Such use includes, but is not limited to, use in life support systems. Buyers of this emulator product must notify the relevant Hitachi sales offices before planning to use the product in such applications.

Improvement Policy:

Hitachi, Ltd. (including its subsidiaries, hereafter collectively referred to as Hitachi) pursues a policy of continuing improvement in design, performance, and safety of the emulator product. Hitachi reserves the right to change, wholly or partially, the specifications, design, user's manual, and other documentation at any time without notice.

Target User of the Emulator Product:

This emulator product should only be used by those who have carefully read and thoroughly understood the information and restrictions contained in the user's manual. Do not attempt to use the emulator product until you fully understand its mechanism.

It is highly recommended that first-time users be instructed by users that are well versed in the operation of the emulator product.

LIMITED WARRANTY

Hitachi warrants its emulator products to be manufactured in accordance with published specifications and free from defects in material and/or workmanship. Hitachi, at its option, will replace any emulator products returned intact to the factory, transportation charges prepaid, which Hitachi, upon inspection, shall determine to be defective in material and/or workmanship. The foregoing shall constitute the sole remedy for any breach of Hitachi's warranty. See the Hitachi warranty booklet for details on the warranty period. This warranty extends only to you, the original Purchaser. It is not transferable to anyone who subsequently purchases the emulator product from you. Hitachi is not liable for any claim made by a third party or made by you for a third party.

DISCLAIMER

HITACHI MAKES NO WARRANTIES, EITHER EXPRESS OR IMPLIED, ORAL OR WRITTEN, EXCEPT AS PROVIDED HEREIN, INCLUDING WITHOUT LIMITATION THEREOF, WARRANTIES AS TO MARKETABILITY, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR USE, OR AGAINST INFRINGEMENT OF ANY PATENT. IN NO EVENT SHALL HITACHI BE LIABLE FOR ANY DIRECT, INCIDENTAL OR CONSEQUENTIAL DAMAGES OF ANY NATURE, OR LOSSES OR EXPENSES RESULTING FROM ANY DEFECTIVE EMULATOR PRODUCT, THE USE OF ANY EMULATOR PRODUCT, OR ITS DOCUMENTATION, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGES. EXCEPT AS EXPRESSLY STATED OTHERWISE IN THIS WARRANTY, THIS EMULATOR PRODUCT IS SOLD "AS IS ", AND YOU MUST ASSUME ALL RISK FOR THE USE AND RESULTS OBTAINED FROM THE EMULATOR PRODUCT.

State Law:

Some states do not allow the exclusion or limitation of implied warranties or liability for incidental or consequential damages, so the above limitation or exclusion may not apply to you. This warranty gives you specific legal rights, and you may have other rights which may vary from state to state.

The Warranty is Void in the Following Cases:

Hitachi shall have no liability or legal responsibility for any problems caused by misuse, abuse, misapplication, neglect, improper handling, installation, repair or modifications of the emulator product without Hitachi's prior written consent or any problems caused by the user system.

All Rights Reserved:

This user's manual and emulator product are copyrighted and all rights are reserved by Hitachi. No part of this user's manual, all or part, may be reproduced or duplicated in any form, in hard-copy or machine-readable form, by any means available without Hitachi's prior written consent.

Other Important Things to Keep in Mind:

1. Circuitry and other examples described herein are meant merely to indicate the characteristics and performance of Hitachi's semiconductor products. Hitachi assumes no responsibility for any intellectual property claims or other problems that may result from applications based on the examples described herein.

2. No license is granted by implication or otherwise under any patents or other rights of any third party or Hitachi.

Figures:

Some figures in this user's manual may show items different from your actual system.

MCU names:

This user’s manual uses H8/xxxx as an example of the MCU names.

Limited Anticipation of Danger:

Hitachi cannot anticipate every possible circumstance that might involve a potential hazard. The warnings in this user's manual and on the emulator product are therefore not all inclusive. Therefore, you must use the emulator product safely at your own risk.

SAFETY PAGE

READ FIRST

• READ this user's manual before using this emulator product.

• KEEP the user's manual handy for future reference.

Do not attempt to use the emulator product until you fully understand its mechanism.

DEFINITION OF SIGNAL WORDS

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

⚠ DANGER **DANGER** indicates an imminently hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING **WARNING** indicates a potentially hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION **CAUTION** indicates a potentially hazardous situation which, if not avoided, may result in minor or moderate injury.

CAUTION **CAUTION** used without the safety alert symbol indicates a potentially hazardous situation which, if not avoided, may result in property damage.

NOTE emphasizes essential information.

⚠ WARNING

Observe the precautions listed below. Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

1. **Do not repair or remodel the emulator product by yourself for electric shock prevention and quality assurance.**
2. **Always switch OFF the host computer and user system before connecting or disconnecting any CABLES or PARTS.**
3. **Connect the connectors in the user system and in the user interface cable by confirming the correct direction.**
4. **If the E10A emulator PCMCIA and PCI cards are mounted on the same host computer, the connectors may be illegally connected.**

Warnings on Emulator Usage

Be sure to read and understand the warnings below before using this emulator. Note that these are the main warnings, not the complete list.

⚠ WARNING

Always switch OFF the host computer and user system before connecting or disconnecting any CABLES or PARTS.

Failure to do so will result in a FIRE HAZARD and will damage the user system and the emulator product or will result in PERSONAL INJURY. The USER PROGRAM will be LOST.

CAUTION

Place the host computer and user system so that no cable is bent or twisted. A bent or twisted cable will impose stress on the user interface leading to connection or contact failure.

Make sure that the host computer and the user system are placed in a secure position so that they do not move during use nor impose stress on the user interface.

CAUTION

This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

Preface

Thank you for purchasing the E10T emulator.

CAUTION

READ section 2, Preparation before Use, of this User's Manual before using the emulator product. Incorrect operation will damage the user system and the emulator product.

This emulator is an efficient development tool for software and hardware of user systems based on Hitachi's original microprocessor. The emulator operates using the Hitachi debugging interface (hereafter referred to as the HDI), which is the interface program that runs on Microsoft® Windows® 98, Microsoft® Windows® ME, Microsoft® Windows NT®, or Microsoft® Windows® 2000 operating system.

This manual describes the functions and operating procedures of the E10T emulator. Sections 1 to 5 describe common features of all types of E10T emulators. Section 6 describes supplements to the E10T emulator.

This manual consists of six sections. The information contained in each section is summarized below:

- Section 1, Overview, gives the emulator overview.
- Section 2, Preparation before Use, gives instructions for first-time users, such as preparation before use and system connection.
- Section 3, Tutorial, describes HDI operating examples.
- Section 4, Descriptions of Windows, describes HDI windows for operating the emulator.
- Section 5, Command-line Functions describes how to input HDI commands and command types.
- Section 6, H8/xxxx E10T Emulator Specifications describes the features of the E10T emulator for each MCU. Section 7 describes the important information of the E10T emulator according to emulator products. Read these sections before using the E10T emulator.

The HDI installation disks are provided by the CD-R. Refer to the descriptions in the manuals of the host computer or operating system.

Related Manuals:

- Supplementary Informations
- Hitachi Debugging Interface User's Manual (HS6400DIIW5SE)
- H8S, H8/300 Series Cross Assembler User's Manual
- H Series Linkage Editor, Librarian, and Object Converter User's Manual
- H8S, H8/300 Series C/C++ Compiler User's Manual
- Hardware Manual for each MCU
- Programming Manual for each MCU

Notes: 1. **IBM PC is a registered trademark of International Business Machines Corporation.**

2. **Microsoft®, Windows®, and Windows NT® are registered trademarks of Microsoft Corporation in the United States and/or other countries.**
   Microsoft® Windows® 98 operating system is referred to as Windows® 98 in this user's manual.
   Microsoft® Windows® Millennium Edition operating system is referred to as Windows® ME in this user's manual.
   Microsoft® Windows NT® operating system is referred to as Windows NT® in this user's manual.
   Microsoft® Windows® 2000 operating system is referred to as Windows® 2000 in this user's manual.

Contents

Section 1 Overview 1
1.1 Environmental Conditions 3
1.2 Components 5

Section 2 Preparation before Use 7
2.1 Emulator Preparation 7
2.2 HDI Installation 8
2.2.1 Installing under Windows® 98 and Windows® ME Operating Systems 8
2.2.2 Installing under Windows NT® 4.0 Operating System 9
2.2.3 Installing under Windows® 2000 Operating System 10
2.3 Connecting the Host Computer with the Card Emulator 11
2.4 Connecting the Card Emulator with the User System 12
2.5 System Check 14
2.5.1 H8/xxxx E10T Emulator Mode 17
2.5.2 Writing H8/xxxx E10T Flash memory Mode 18
2.6 Ending the HDI 22
2.7 Uninstalling the HDI 23
2.8 CD-R 24
2.8.1 Configuration of the CD-R 24

Section 3 Tutorial 25
3.1 Introduction 25
3.2 Running the HDI 26
3.3 [HDI] Window 27
3.4 Setting up the Emulator 28
3.5 Setting the [Configuration] Dialog Box 28
3.6 Downloading the Tutorial Program 30
3.6.1 Downloading the Tutorial Program 30
3.6.2 Displaying the Source Program 31
3.7 Setting the Software Breakpoint 33
3.8 Setting Registers 34
3.9 Executing the Program 36
3.10 Reviewing Breakpoints 39
3.11 Viewing Memory 40
3.12 Watching Variables 41
3.13 Stepping Through a Program 44
3.13.1 Executing [Step In] Command 45
3.13.2 Executing [Step Out] Command 46
3.13.3 Executing [Step Over] Command 48

3.14 Displaying Local Variables 50
3.15 Break Function 51
3.15.1 Software Break Function 51
3.16 Hardware Break Function 58
3.17 Trace Function 65
3.18 What Next? 66

Section 4 Descriptions of Windows 67
4.1 HDI Windows 67
4.2 Descriptions of Each Window 70
4.2.1 [Configuration] Dialog Box 70
4.2.2 [E10T Driver Details] Dialog Box 76
4.2.3 [Breakpoints] Window 78
4.2.4 [Break] Dialog Box 80
4.2.5 [Breakpoint] Dialog Box 85
4.2.6 [Break condition] Dialog Box 87
4.2.7 [Break condition] Dialog Box Pages 89
4.2.8 [Trace] Window 93
4.2.9 [System Status] Window 95

Section 5 Command-line Functions 97
5.1 Table and Symbol Description 97
5.1.1 Format 97
5.1.2 Parameter Input 97
5.1.3 Examples 98
5.1.4 Related Items 98
5.2 Command Descriptions 99
5.2.1 BREAKCONDITION_CLEAR: BCC 100
5.2.2 BREAKCONDITION_DISPLAY: BCD 101
5.2.3 BREAKCONDITION_ENABLE: BCE 102
5.2.4 BREAKCONDITION_SET: BCS 103
5.2.5 BREAKPOINT: BP 106
5.2.6 BREAKPOINT_CLEAR: BC 107
5.2.7 BREAKPOINT_DISPLAY: BD 108
5.2.8 BREAKPOINT_ENABLE: BE 109
5.2.9 DEVICE_TYPE: DE 110
5.2.10 GO_OPTION: GP 111
5.2.11 REFRESH: RF 112
5.2.12 RESET: RE 113
5.2.13 STATUS: STS 114
5.2.14 STEP_INTERRUPT: SI 115
5.2.15 TRACE_DISPLAY: TD 116

Section 6 H8/3048F-ONE E10T Emulator Specifications 117
6.1 Components of the Emulator 117
6.2 Pin Arrangement of the Connector 118
6.3 Differences between the H8/3048F-ONE and the Emulator 123
6.4 The H8/3048F-ONE E10T Emulator Functions 124
6.4.1 Emulator Driver Selection 124
6.4.2 Hardware Break Functions 125
6.4.3 Notes on Setting the [Breakpoint] Dialog Box 126
6.4.4 Trace Function 126
6.4.5 Notes on HDI 127

Figures

Figure 1.1 System Configuration with the Emulator (PCMCIA Card Emulator Used) 1
Figure 1.2 System Configuration with the Emulator (PCI Card Emulator Used) 2
Figure 2.1 Emulator Preparation Flow Chart 7
Figure 2.2 Inserting the PCMCIA Card Emulator into the Host Computer 11
Figure 2.3 Inserting the PCI Card Emulator into the Host Computer 11
Figure 2.4 Connecting the User System Interface Cable to the User System 12
Figure 2.5 [Start] Menu 14
Figure 2.6 [Select Session] Dialog Box 15
Figure 2.7 [E10T Driver Details] Dialog Box 16
Figure 2.8 Dialog Box of the Power-on Request Message 17
Figure 2.9 [System Clock] Dialog Box 17
Figure 2.10 [ID Code] Dialog Box 18
Figure 2.11 [HDI] Status Bar 18
Figure 2.12 [System Clock] Dialog Box 18
Figure 2.13 [Load Program] Dialog Box 19
Figure 2.14 [Continue?] Window 19
Figure 2.15 [Connector disconnected] Dialog Box 19
Figure 2.16 [Flash memory erase error!] Dialog Box 20
Figure 2.17 [Error sending firmware & user program] Dialog Box 20
Figure 2.18 [ID code error!] Dialog Box 20
Figure 2.19 [Boot Failed!] Dialog Box 21
Figure 2.20 [Unable to restore the previous driver settings] Dialog Box 21
Figure 2.21 [Exit HDI] Dialog Box 22
Figure 2.22 [Save session] Dialog Box 22
Figure 3.1 [Start] Menu 26
Figure 3.2 [HDI] Window 27
Figure 3.3 [Configuration] Dialog Box 28
Figure 3.4 [Load Program] Dialog Box 30
Figure 3.5 [HDI] Dialog Box 30
Figure 3.6 [Open] Dialog Box 31
Figure 3.7 [Program] Window (Displaying the Source Program) 32
Figure 3.8 [Program] Window (Setting a Software Breakpoint) 33
Figure 3.9 [Registers] Window 34
Figure 3.10 [Register] Dialog Box (PC) 35
Figure 3.11 [Register] Dialog Box (ER7) 35
Figure 3.12 [Go] Button 36
Figure 3.13 [Reset Go] Button 36
Figure 3.14 [Program] Window (Break Status) 36
Figure 3.15 [System Status] Window 37
Figure 3.16 [Breakpoints] Window 39
Figure 3.17 [Open Memory Window] Dialog Box 40

Figure 3.18 [Memory] Window 40
Figure 3.19 [Instant Watch] Dialog Box 41
Figure 3.20 [Watch] Window (Displaying the Array) 42
Figure 3.21 [Add Watch] Dialog Box 42
Figure 3.22 [Watch] Window (Displaying the Variable) 43
Figure 3.23 [Watch] Window (Displaying Array Elements) 43
Figure 3.24 [Step In] Button 45
Figure 3.25 [Program] Window (Step In) 45
Figure 3.26 [Step Out] Button 46
Figure 3.27 [Program] Window (Step Out) 46
Figure 3.28 [Program] Window (Step In –> Step In) 47
Figure 3.29 [Program] Window (Before Step Over Execution) 48
Figure 3.30 [Step Over] Button 48
Figure 3.31 [Program] Window (Step Over) 49
Figure 3.32 [Locals] Window 50
Figure 3.33 [Breakpoints] Window (Before Software Breakpoint Setting) 51
Figure 3.34 [Point] Page ([Break] Dialog Box) 52
Figure 3.35 [Breakpoint] Dialog Box 53
Figure 3.36 [Point] Page ([Break] Dialog Box) (After Software Breakpoint Setting) 54
Figure 3.37 [Breakpoints] Window (Software Breakpoint Setting) 55
Figure 3.38 [Program] Window at Execution Stop (Software Break) 56
Figure 3.39 Displayed Contents of the [System Status] Window (Software Break) 57
Figure 3.40 [Breakpoints] Window (Before Hardware Break Condition Setting) 58
Figure 3.41 [Condition] Page ([Break] Dialog Box) 59
Figure 3.42 [condition] Page ([Break condition 1] Dialog Box) 60
Figure 3.43 [Break] Dialog Box (After Hardware Break Condition Setting) 61
Figure 3.44 [Breakpoints] Window ([Break condition 1] Setting) 62
Figure 3.45 [Program] Window at Execution Stop (Break condition 1) 63
Figure 3.46 Displayed Contents of the [System Status] Window (Break condition 1) 64
Figure 3.47 [Trace] Window 65
Figure 4.1 [Configuration] Dialog Box 70
Figure 4.2 [General] Page ([Configuration] Dialog Box) 72
Figure 4.3 Warning Message Box 74
Figure 4.4 [E10T Driver Details] Dialog Box 76
Figure 4.5 [Breakpoints] Window 78
Figure 4.6 [Break] Dialog Box 80
Figure 4.7 [Point] Page ([Break] Dialog Box) 81
Figure 4.8 [Condition] Page ([Break] Dialog Box) 83
Figure 4.9 [Breakpoint] Dialog Box 85
Figure 4.10 [Break condition 1] Dialog Box 87
Figure 4.11 [condition] Page 90
Figure 4.12 [Trace] Window 93
Figure 4.13 [System Status] Window 95

Figure 6.1 Pin Arrangement of the Connector 118
Figure 6.2 Connection of the Emulator (Mode 7) 119
Figure 6.3 Connection of Emulator and P91, P93, and P95 Pin 119
Figure 6.4 Direct Connection of Emulator and FWE Pin 120
Figure 6.5 Example of Reset Circuits 120
Figure 6.6 Example of Circuits for Operating Mode Setting Pins 121
Figure 6.7 Interface Circuit in the Emulator (Reference) 122

Tables

Table 1.1 Environmental Conditions 3
Table 1.2 Operating Environments 4
Table 2.1 Recommended Connector 12
Table 2.2 [E10T Driver Details] Dialog Box Option 16
Table 3.1 Tutorial Program: Configuration and Parts 25
Table 3.2 Setting the [Configuration] Dialog Box 29
Table 3.3 Contents of the [System Status] Window 38
Table 3.4 Step Option 44
Table 4.1 HDI Window Menus and Related Manual Entries 67
Table 4.2 [Configuration] Dialog Box Page 71
Table 4.3 [General] Page Options 73
Table 4.4 [E10T Driver Details] Dialog Box Option 76
Table 4.5 [Breakpoints] Window Display Items 78
Table 4.6 [Breakpoints] Window Pop-up Menu Operation 79
Table 4.7 [Break] Dialog Box Pages 80
Table 4.8 [Point] Page Options 82
Table 4.9 [Condition] Page Options 84
Table 4.10 [Address] Page Options 86
Table 4.11 Setting Conditions in [Break condition] Dialog Box 89
Table 4.12 [Break condition] Dialog Box Pages 89
Table 4.13 [Address] Group Box Options 91
Table 4.14 Radio Button Options 91
Table 4.15 [Data] Group Box Options 91
Table 4.16 [Read/Write] Group Box Options 92
Table 4.17 [Trace] Window Display Items 94
Table 4.18 [System Status] Window Display Items 96
Table 5.1 E10T HDI Commands 99
Table 5.2 BREAKCONDITION_CLEAR Command Parameter 100
Table 5.3 BREAKCONDITION_DISPLAY Command Parameter 101
Table 5.4 BREAKCONDITION_ENABLE Command Parameters 102
Table 5.5 BREAKCONDITION_SET Command Parameters 104
Table 5.6 BREAKPOINT Command Parameters 106
Table 5.7 BREAKPOINT_CLEAR Command Parameters 107
Table 5.8 BREAKPOINT_DISPLAY Command Parameter 108
Table 5.9 BREAKPOINT_ENABLE Command Parameters 109
Table 5.10 DEVICE_TYPE Command Parameter 110
Table 5.11 GO_OPTION Command Parameter 111
Table 5.12 REFRESH Command Parameter 112
Table 5.13 RESET Command Parameter 113
Table 5.14 STATUS Command Parameter 114
Table 5.15 STEP_INTERRUPT Command Parameter 115

Table 5.16 TRACE_DISPLAY Command Parameter 116
Table 6.1 Components of the Emulator (HS3048BTCM01H or HS3048BTCI01H) 117
Table 6.2 Register Initial Values at Emulator Power-On 123
Table 6.3 Type Name and Driver 124
Table 6.4 Hardware Break Condition Specification Items 125
Table 6.5 Conditions Set in [Break condition 1] and [Break condition 2] Dialog Boxes 125
Table 6.6 Conditions Set by BREAKCONDITION_SET Command 125

Section 1 Overview

The E10T emulator (hereafter referred to as the emulator) is a software and hardware development support tool for application systems using the microprocessor developed by Hitachi, Ltd.

The PCMCIA card emulator or PCI card emulator (hereafter referred to as the card emulator), which is the main unit of the emulator, is connected, through the dedicated debugging interface, to the user system. The user system can be debugged under the conditions similar to the actual application conditions. The emulator enables debugging anywhere indoors or out. The host computer for controlling the emulator must be an IBM PC compatible machine with a PCMCIA type II or PCI slot.

Figures 1.1 and 1.2 show the system configuration using the emulator.

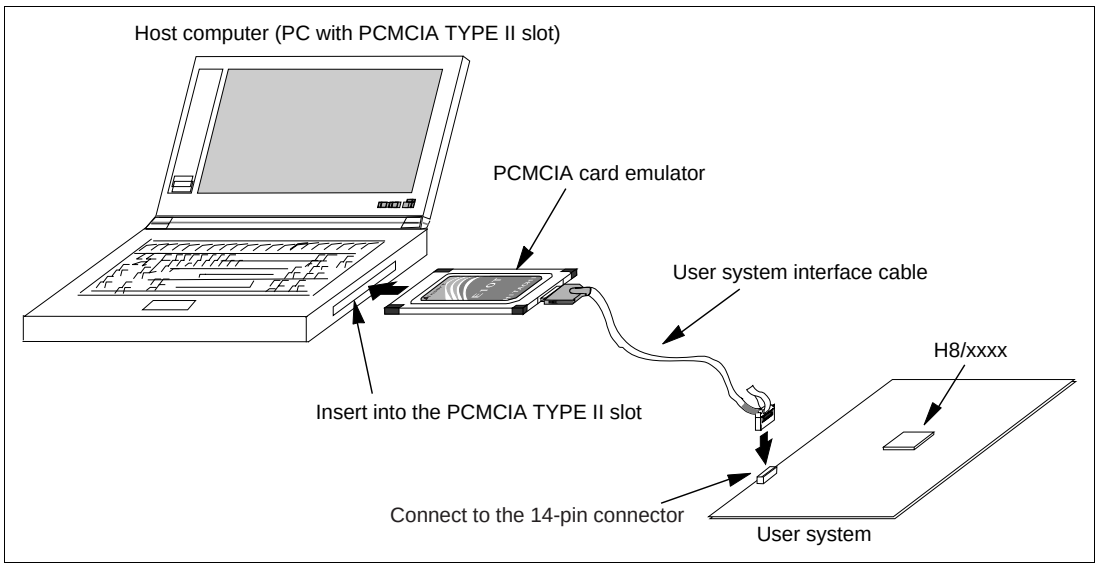

Host computer (PC with PCMCIA TYPE II slot)
PCMCIA card emulator
User system interface cable
H8/xxxx
Insert into the PCMCIA TYPE II slot
Connect to the 14-pin connector
User system

Figure 1.1 System Configuration with the Emulator (PCMCIA Card Emulator Used)

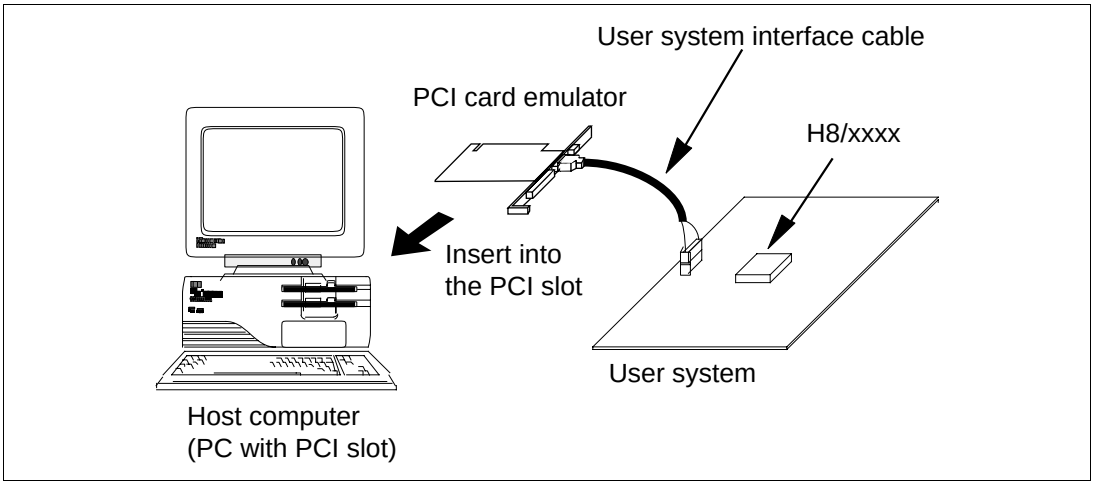


Figure 1.2 System Configuration with the Emulator (PCI Card Emulator Used)

The emulator has three activation modes:

1. A mode in which the emulator program is linked up after it has been downloaded to the target device:

   This mode is used when there is no emulator program in the flash memory of the target device. In addition, the user program can be downloaded at the same time to reduce the number of times that the flash memory is rewritten.
2. A mode in which the emulator program is linked up without downloading it to the target device:

   This mode is used when there is an emulator program in the flash memory of the target device.
3. A mode in which the emulator is used as the flash memory writer:

   This mode is used when the emulator is used for writing to flash memory.

The emulator provides the following features:

- Excellent cost-performance card emulator

  Compactness and low price are implemented using the PCMCIA interface or the PCI interface.
- Realtime emulation

  Realtime emulation of the user system is enabled at the maximum operating frequency of the MCU.
- Excellent operability

  Using the Hitachi Debugging Interface (HDI) on the Microsoft® Windows® 98, Microsoft® Windows® ME, Microsoft® Windows NT®, and Microsoft® Windows® 2000 operating systems enables user program debugging using a pointing device such as a mouse. The HDI enables high-speed downloading of load module files.

- Various debugging functions

  Various break and trace functions enable efficient debugging. Breakpoints and break conditions can be set by the specific window, trace information can be displayed on a window, and command-line functions can be used.
- Memory access during emulation

  During emulation, the memory contents can be read and modified.
- Debugging of the user system in the final development stage

  The user system can be debugged under conditions similar to the actual application conditions.
- Compact debugging environment

  When the card emulator specific to the PCMCIA interface is used, a laptop computer can be used as a host computer, creating a debugging environment in any place.
- The emulator can be used to write to the internal flash memory.

1.1 Environmental Conditions

CAUTION

Observe the conditions listed in tables 1.1 and 1.2 when using the emulator. Failure to do so will damage the user system and the emulator product. The USER PROGRAM will be LOST.

Table 1.1 Environmental Conditions

Item	Specifications
Temperature	Operating: +10°C to +35°C Storage: –10°C to +50°C
Humidity	Operating: 35% RH to 80% RH, no condensation Storage: 35% RH to 80% RH, no condensation
Vibration	Operating: 2.45 m/s^2 max. Storage: 4.9 m/s^2 max. Transportation: 14.7 m/s^2 max.
Ambient gases	There must be no corrosive gases present

Table 1.2 lists the acceptable operating environments.

Table 1.2 Operating Environments

Item	Description
Host computer	Built-in Pentium or higher-performance CPU (166 MHz or higher recommended); IBM PC or compatible machine with the PCMCIA TYPE II slot or the PCI slot.
OS	Windows® 98, Windows® ME, Windows NT®, and Windows® 2000
Minimum memory capacity	32 Mbytes or more (double of the load module size recommended)
Hard-disk capacity	Installation disk capacity: 10 Mbytes or more. (Prepare an area at least double the memory capacity (four-times or more recommended) as the swap area.)
Pointing device such as mouse	Connectable to the host computer; compatible with Windows® 98, Windows® ME, Windows NT®, and Windows® 2000.
Power voltage	5.0 ± 0.25 V (The power is supplied to the emulator by the host computer.)
Current consumption	HS0005TCM02H: 55 mA (max) HS0005TCI02H: 55 mA (max)
CD-ROM drive	Required to install the emulator or refer to the emulator user's manual.

1.2 Components

Check all the components unpacking. For details on the E10T emulator components, refer to section 6.1, Components of the Emulator. If the components are not complete, contact a Hitachi sales agency.

2.1 Emulator Preparation

⚠ WARNING

READ the reference sections shaded in figure 2.1 before using the emulator product. Incorrect operation will damage the user system and the emulator product. The USER PROGRAM will be LOST.

Unpack the emulator and prepare it for use as follows:

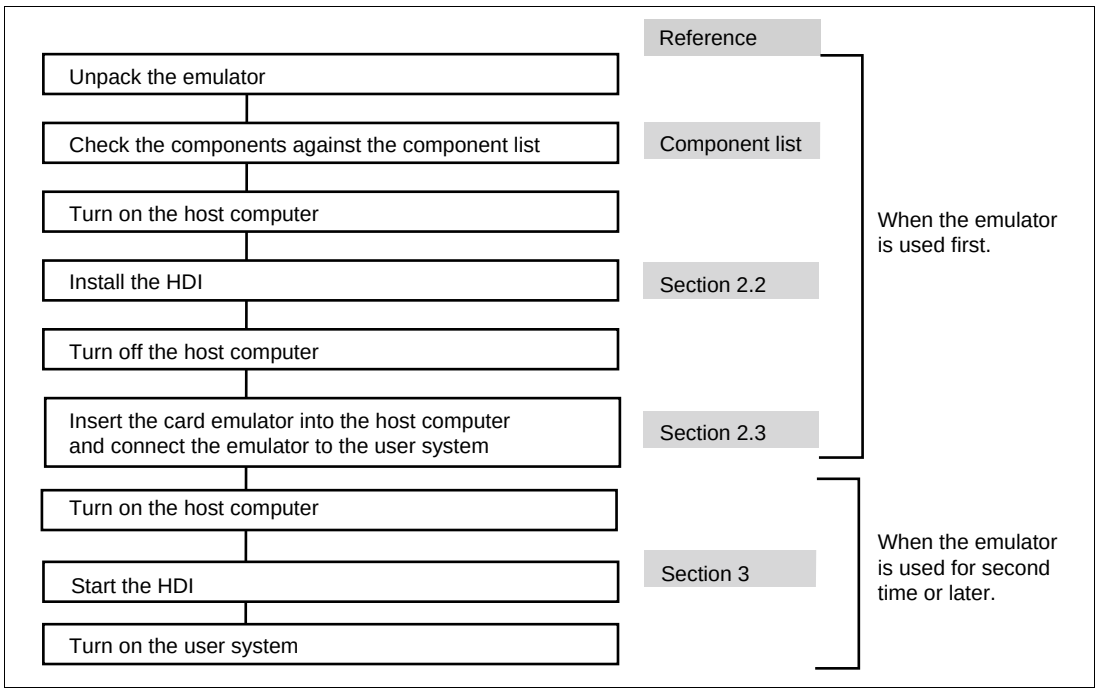

Reference
Unpack the emulator
Check the components against the component list
Component list
Turn on the host computer
Install the HDI
Section 2.2
Turn off the host computer
Insert the card emulator into the host computer and connect the emulator to the user system
Section 2.3
When the emulator is used first.
Turn on the host computer
Start the HDI
Section 3
Turn on the user system
When the emulator is used for second time or later.

Figure 2.1 Emulator Preparation Flow Chart

2.2 HDI Installation

When the CD-R is inserted in the host computer's CD-ROM drive, the HDI installation wizard is automatically activated (holding the Shift key down while the CD-R is inserted cancels this automatic activation). To run the installation wizard when it has not been automatically activated, execute Setup.exe from the root directory of the CD-R.

Follow the cues given by the installation wizard to install the HDI.

Since hardware settings are also made during installation, the installation procedure differs according to the operating system or interface (PCI or PCMCIA) being used. Follow the installation steps carefully according to the environment you are using.

2.2.1 Installing under Windows® 98 and Windows® ME Operating Systems

(1) When the emulator is a PCI card:

1. Install the HDI (when the component type has to be selected during installation, be sure to select [PCI Card Driver]).
2. Shut the operating system down and turn off the power to the host computer.
3. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
4. Restart the host computer. The hardware is now recognized and the driver is automatically installed.*

(2) When the emulator is a PCMCIA card:

1. Install the HDI (when the component type has to be selected during installation, be sure to select [PC Card Driver (PCMCIA)]).
2. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
3. The hardware is now recognized and the driver is automatically installed.*

Note: When [Add New Hardware Wizard] is displayed, select the [Search for the best driver for your device. (Recommended)] radio button and then the [Specify a location] check box to select the path to be searched for drivers. The location must be specified according to the emulator type, as indicated below:
When using the PCI-card emulator: <Drive>:\DRIVERS\PCI\95
When using the PCMCIA-card emulator: <Drive>:\DRIVERS\PCMCIA\95
(<Drive> is the CD-ROM drive name.)

2.2.2 Installing under Windows NT® 4.0 Operating System

(1) When the emulator is a PCI card:

1. Shut the operating system down and turn off the power to the host computer.
2. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
3. Start the host computer and log-on with an administrator-level user name.
4. Install the HDI. (For a component, be sure to select [PCI Card Driver]. There is a check box for selecting the type name of the product under the [PCI Card Driver] component. Select the appropriate type name. If the correct name is not selected, the correct driver will not be installed, and the emulator will not operate.)
5. Restart the host computer.

(2) When the emulator is a PCMCIA card:

1. Shut the operating system down and turn off the power to the host computer.
2. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
3. Start the host computer and log-on with an administrator-level user name.
4. During HDI installation, the setting value should be checked beforehand because inquiries are made about the resource used by the PCMCIA-card emulator.

   Start the [Start] menu -> [Programs] -> [Administrative Tools (Common)] -> [Windows NT Diagnostics], check the status of the IRQ, I/O port, and memory from the resource panel, and determine the setting values that do not conflict with other devices. (The following resources are used: IRQ: one channel, I/O port: H'F byte, and memory: H'4000 byte.)
5. Install the HDI. (For a component, be sure to select [PC Card Driver (PCMCIA)]. There is a check box for selecting the type name of each product under the [PC Card Driver (PCMCIA)] component. Select the appropriate type name. If the correct name is not selected, the correct driver will not be installed and the emulator will not operate.)
6. Restart the host computer.

Notes: The driver that has been selected in the [Drivers] component starts after the host computer is initiated. If the host computer is initiated with the card disconnected or with the incorrect driver installed, the driver cannot initiate and the service control manager informs the system of an error. This, however, is not a problem.

2.2.3 Installing under Windows® 2000 Operating System

(1) When the emulator is a PCI card:

1. Log-on with an administrator-level user name.
2. Install the HDI. (When a component is selected, be sure to select [PCI Card Driver].)
3. Shut the operating system down and turn off the power to the host computer.
4. Insert the PCI-card emulator in a slot on the host computer. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
5. Restart the host computer and log-on with an administrator-level user name. The hardware is now recognized and the driver is automatically installed.*

(2) When the emulator is a PCMCIA card:

1. Log-on with an administrator-level user name.
2. Install the HDI. (When a component is selected, be sure to select [PC Card Driver (PCMCIA)].)
3. Insert the PCMCIA-card emulator in the host computer's slot. Refer to section 2.3, Connecting the Host Computer with the Card Emulator.
4. The hardware is now recognized and the driver is automatically installed.*

Note: When [Found New Hardware Wizard] is displayed, select the [Search for a suitable driver for my device (recommended).] radio button and then the [Specify a location] check box to select the path to be searched for drivers. The location must be specified according to the emulator type, as indicated below:
When using the PCI-card emulator: <Drive>:\DRIVERS\PCI\2000
When using the PCMCIA-card emulator: <Drive>:\DRIVERS\PCMCIA\2000
(<Drive> is the CD-ROM drive name.)

2.3 Connecting the Host Computer with the Card Emulator

Insert the card emulator into the PCMCIA TYPE II slot or the PCI slot of the host computer (figures 2.2 and 2.3).

Note: Be sure to install the HDI before the card emulator is inserted.

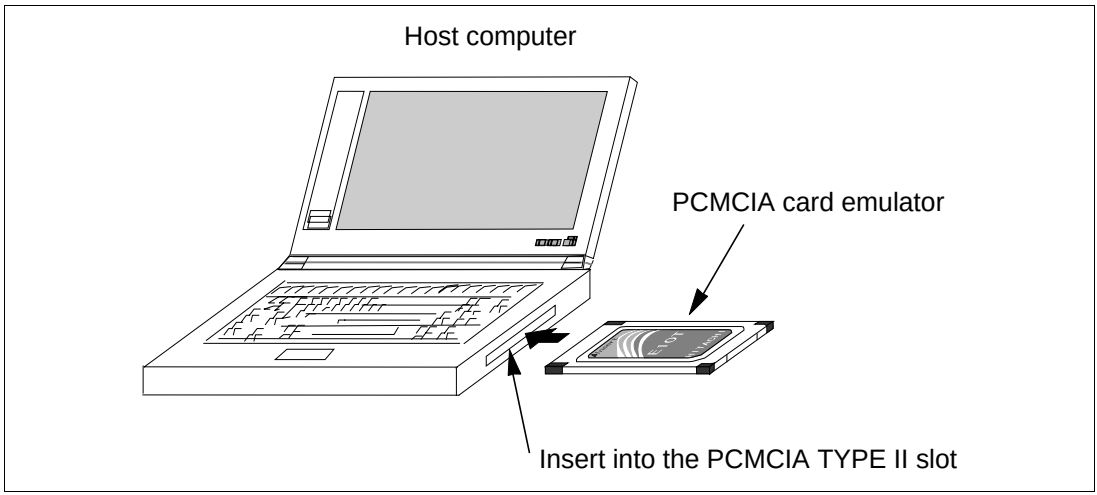


Figure 2.2 Inserting the PCMCIA Card Emulator into the Host Computer

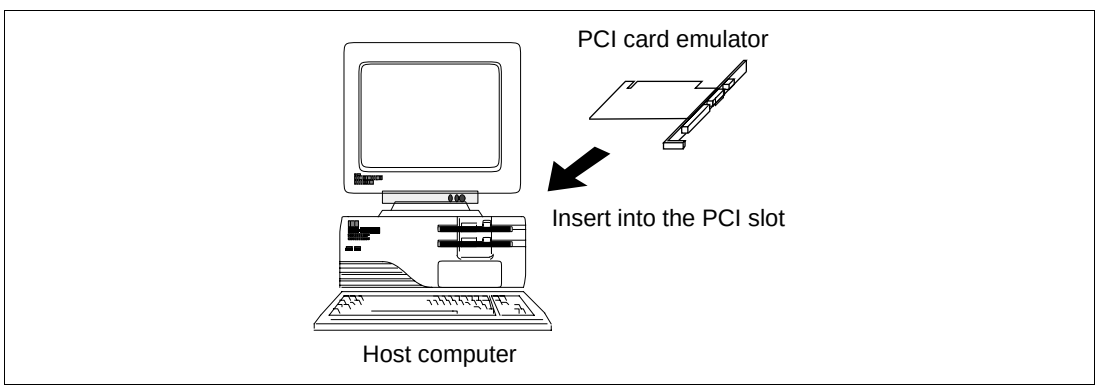


Figure 2.3 Inserting the PCI Card Emulator into the Host Computer

Use the procedure, described in section 2.4, to connect the emulator to the user system with the user system interface cable, or to disconnect them when moving the emulator or the user system.

Note: When installing the PCI card emulator, note the following:

1. Turn off the host computer.
2. Insert the emulator into the PCI slot in parallel.
3. Screw in the emulator after confirming the connector and cable positions.

2.4 Connecting the Card Emulator with the User System

(1) The connector must be installed to the user system. Table 2.1 shows the recommended connector for the emulator.

Table 2.1 Recommended Connector

Type Number	Manufacturer	Specifications
2514-6002	3M Limited	20-pin straight type

Note: When the connector is used, do not install any components within 3 mm of the connector.

(2) The pin arrangement of the connector is shown in section 6.2, Pin Arrangement of the Connector.

(3) Figure 2.4 shows how to connect the user system interface cable to the user system. Connect the ground line of the cable to the user system ground. The end of the ground line has a hole having a diameter of 3 mm, and therefore, when the ground line is screwed to the user system, the screw diameter must be 3 mm.

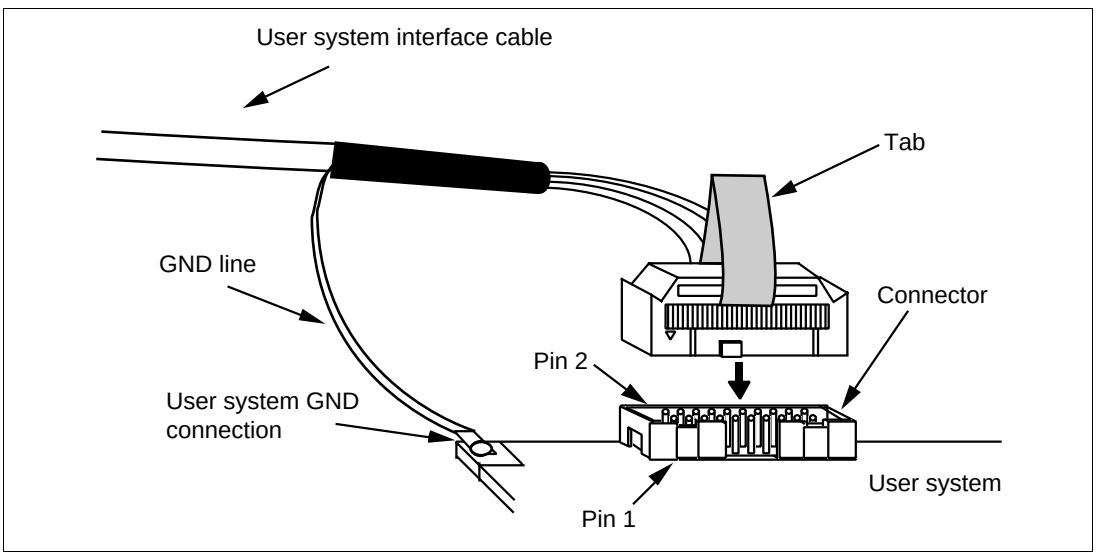


Figure 2.4 Connecting the User System Interface Cable to the User System

Notes:

1. **To connect the signals output from the connector, refer to the MCU pin alignment.**
2. **To remove the user system interface cable from the user system, pull the tab on the connector upward.**
3. **The range of frequencies that the emulator operates at is different according to the MCUs used.**
4. **Connect the signals from the connector as shown in section 6.2, Pin Arrangement of the Connector.**

2.5 System Check

When the HDI program is executed, check that the emulator operates correctly according to the following procedure:

1. Check that the card emulator is inserted into the host computer.
2. Connect the user system interface cable to the connector of the card emulator.
3. Connect the user system interface cable to the connector.
4. Power on the host computer and select [HDI for E10T H8xxxx] from the [Start] menu.

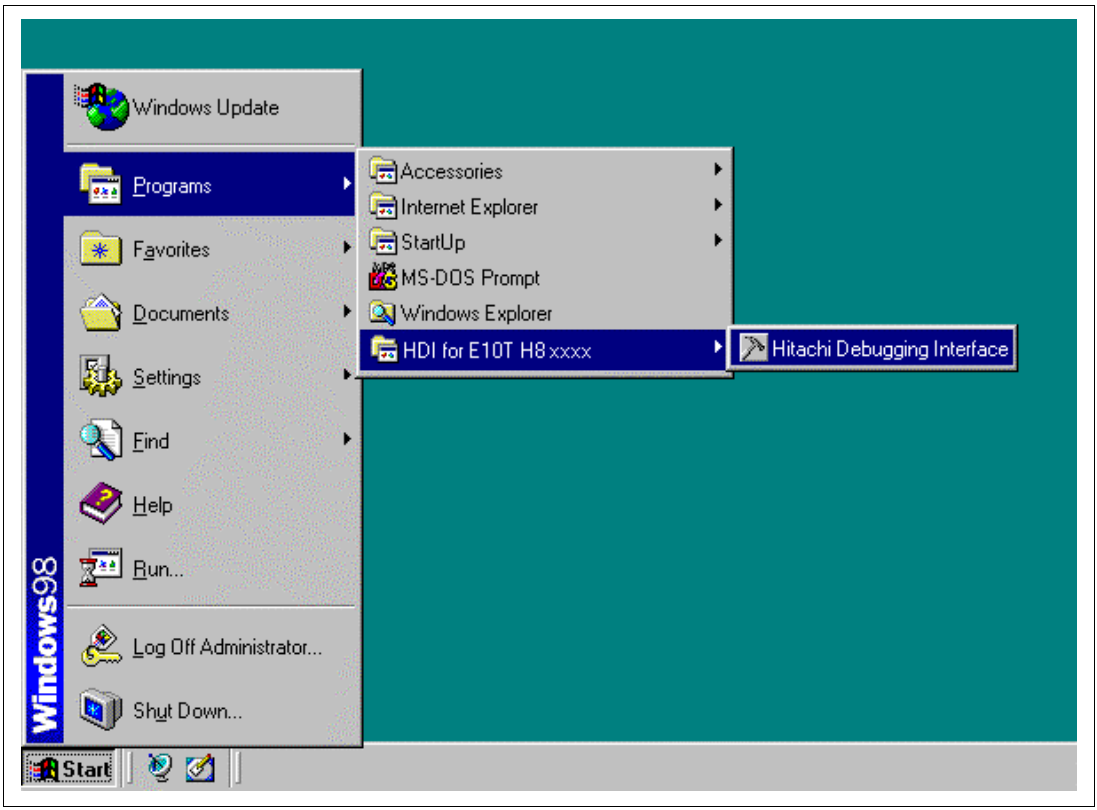


Figure 2.5 [Start] Menu

5. The [Select Session] dialog box is displayed. One of the three can be selected for the session. Select the activation mode for the emulator.

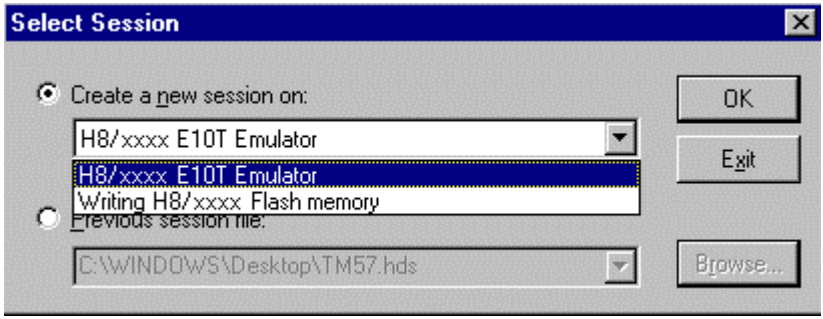


Figure 2.6 [Select Session] Dialog Box

6. The [E10T Driver Details] dialog box is displayed. With the [Driver] combo box, select the driver to connect the HDI with the emulator. [Interface] displays the interface name of the PC interface board to be connected, and [Channel] displays the interface to which the board is connected. Once the driver is selected in the [E10T Driver Details] dialog box, this dialog box is not displayed when the HDI is run next time. (This procedure will not be executed by target MCUs.)

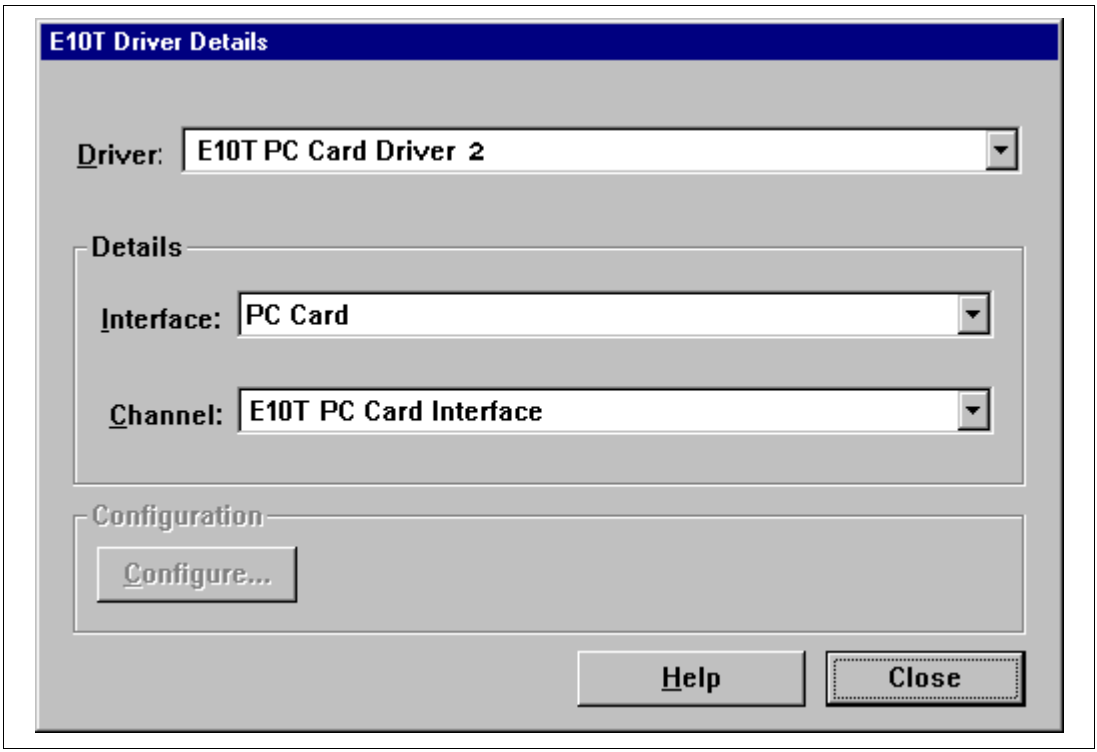


Figure 2.7 [E10T Driver Details] Dialog Box

Table 2.2 lists the options.

Table 2.2 [E10T Driver Details] Dialog Box Option

Option	Description
[Driver] combo box	Selects the driver to be used to connect the HDI with the emulator.
[Interface] combo box	Selects the interface name for connection of the card emulator.
[Channel] combo box	Selects the interface to which the PC interface board has been connected.
[Configure...] button	Displays a configuration dialog box, if the driver supports one.

Select [E10T PC Card Driver 2] when using the PCMCIA card emulator, and [E10T PCI Card Driver 2] when using the PCI card emulator.

7. The HDI window is displayed, and the dialog box is displayed as shown in figure 2.8.

Figure 2.8 Dialog Box of the Power-on Request Message

8. Supply power to the user system.
   The subsequent procedures depend on the activation mode that was selected in step 5.

2.5.1 H8/xxxx E10T Emulator Mode

This mode is used when debugging is performed by the emulator. The following procedures apply when [H8/xxxx E10T Emulator] has been selected from the three activation modes listed in the [Select Session] dialog box.

Note: Only download the user program to the flash memory area. If a program taking up memory that includes RAM is downloaded, a [COMMUNICATION TIMEOUT ERROR] will occur.

1. After the [System Clock] window appears, input the system clock frequency. This frequency value is used when erasing flash memory in the emulator.

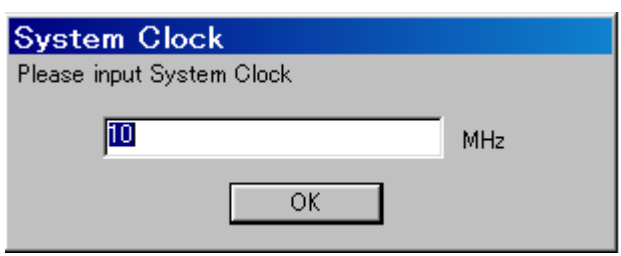


Figure 2.9 [System Clock] Dialog Box

2. Set a hexadecimal eight-digit numerical ID code as a security code for the flash memory. Uncheck the [New registration] check box and input this ID code when [H8/xxxx E10T Emulator] is selected and the HDI is activated in the link-up mode without downloading emulator program to the target device. If the ID code is not matched, the flash memory contents are erased.

Figure 2.10 [ID Code] Dialog Box

3. When "Link up" appears on the status bar, the user and emulator programs have been downloaded to the flash memory, and the HDI initialization is complete.

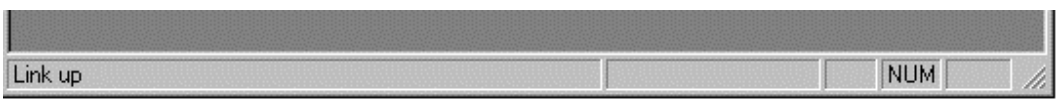


Figure 2.11 [HDI] Status Bar

2.5.2 Writing H8/xxxx E10T Flash memory Mode

In this mode the emulator is used as a flash memory writer. The following procedures apply when [Writing H8/xxxx E10T Flash memory] is selected from the three activation modes listed in the [Select Session] dialog box.

1. After the [System Clock] window appears, input the system clock frequency.

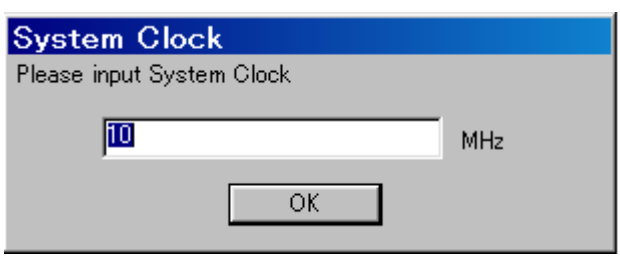


Figure 2.12 [System Clock] Dialog Box

2. When the [Load Program] window appears, specify a user program to be downloaded. Click the [Open] button to start downloading.

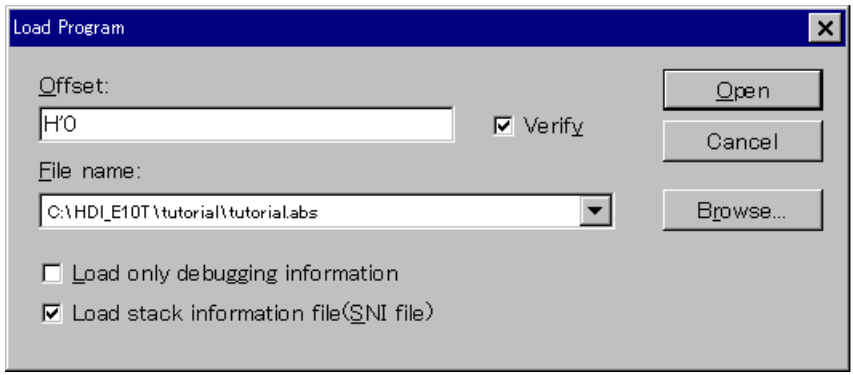


Figure 2.13 [Load Program] Dialog Box

3. When the program has been downloaded, the [Continue] window appears. When the [OK] button is clicked, a message is displayed to request that power be supplied. Turn off the power, exchange the MCU, and supply power. Repeat operations 1 to 3 until the [Exit] button is clicked. When the [Exit] button is clicked, the HDI is terminated.

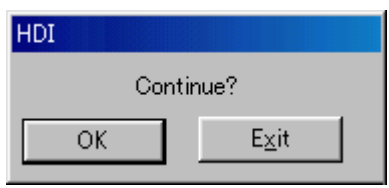


Figure 2.14 [Continue?] Window

Notes: 1. When the HDI does not link up, and the above procedures have been executed, the driver setting must be incorrect. Install a driver from the /SETUP directory of the CD-R according to the screen instructions.

2. If the user system interface cable is disconnected from the connector on the user system, the following dialog box will appear.

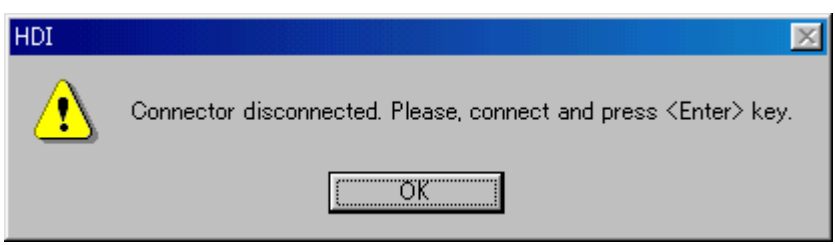


Figure 2.15 [Connector disconnected] Dialog Box

3. If the emulator is not properly initialized, one of the dialog boxes shown in figures 2.16 through 2.19 will appear.

(a) The following dialog box is displayed when the flash memory cannot be erased. Exchange the MCU since the flash memory has been rewritten to more times than the guaranteed value.

Figure 2.16 [Flash memory erase error!] Dialog Box

(b) The following dialog box is displayed when the flash memory cannot be written to. An incorrect CPU operating frequency has been input or the flash memory has been rewritten to more times than the guaranteed value.

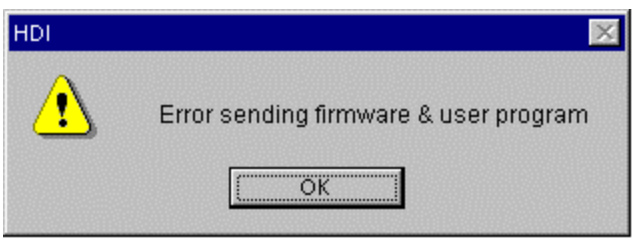


Figure 2.17 [Error sending firmware & user program] Dialog Box

(c) The following dialog box is displayed when an incorrect ID code has been input. For security, the flash memory is completely erased.

Figure 2.18 [ID code error!] Dialog Box

(d) The following dialog box is displayed when the MCU cannot communicate with the emulator. Check the MCU settings.

Figure 2.19 [Boot Failed!] Dialog Box

4. If an incorrect driver has been selected, the following dialog box will appear.

Figure 2.20 [Unable to restore the previous driver settings] Dialog Box

The [E10T Driver Details] dialog box appears after the [OK] button is clicked. Select the correct driver. For details, refer to section 6.4.1, Emulator Driver Selection.

2.6 Ending the HDI

Exit the HDI by using the following procedure:

1. Select [Exit] from the [File] menu to end the HDI. When the [Exit HDI] dialog box is displayed, click the [Yes] button.

Figure 2.21 [Exit HDI] Dialog Box

2. Then, the [Save session] dialog box is displayed. If necessary, click the [Yes] button to save session. After saving session, the HDI ends. If not necessary, click the [No] button to end the HDI.

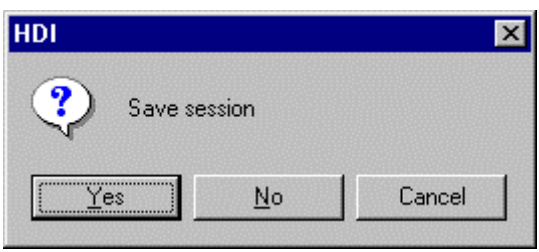


Figure 2.22 [Save session] Dialog Box

3. Turn the user system off.

2.7 Uninstalling the HDI

Follow this procedure to remove the installed HDI from the user's host computer.

1. Open [Add/Remove Programs Properties] from the control panel. Select the HDI program from the list and click the [Add/Remove…] button.
2. The setup program is executed again and the installed application can be changed, modified, or removed. When the application is to be uninstalled, select removal.

CAUTION

A shared file may be detected while the program is being removed. If another HDI may be using the shared file, do not remove the file. When Microsoft® Windows NT®4.0 operating system is used, the removal of the registry information on the driver may be asked. If other HDI may use the target driver, do not remove the registry information. If another HDI does not start up after the removal process, re-install that HDI.

2.8 CD-R

2.8.1 Configuration of the CD-R

The root directory of the CD-R contains a setup program for HDI installation. The folders contain the files and programs listed below.

Table 2.3 Contents of the CD-R Directories

Directory Name	Contents	Description
Dlls	Microsoft® runtime library	A runtime library for the HDI. The version is checked at installation and this library is copied to the hard disk as part of the installation process.
Drivers	E10T emulator driver	The E10T emulator drivers.
Help	Online help for the E10T emulator	An online help file. This is copied to the hard disk as part of the installation process.
Manual	E10T emulator manual	Precautions on Using the E10T Emulator and the user's manual. These are provided as PDF files.
Pdf_read	Adobe® Acrobat® Reader setup program	Adobe® Acrobat® Reader is an application for displaying, viewing, and printing PDF files.

Section 3 Tutorial

3.1 Introduction

The following describes the main functions of the emulator by using a tutorial program.

The tutorial program is based on the C program that sorts ten random data items in ascending or descending order. The tutorial program performs the following actions:

- The `main` function generates random data to be sorted.
- The `sort` function sorts the generated random data in ascending order.
- The `change` function then sorts the data in descending order.

The tutorial program is included in the `sort.c` file. The compiled load module is provided in the Sysrof format and is included in the `tutorial.abs` file.

Table 3.1 lists the tutorial program configuration.

Table 3.1 Tutorial Program: Configuration and Parts

Item	Contents
Workspace for HEW V1.2	[Installation directory]\tutorial\tutorial.hws
Load module	[Installation directory]\tutorial\tutorial\Debug\tutorial.abs
Main program (source file)	[Installation directory]\tutorial\tutorial\tutorial.c
Stack information file	[Installation directory]\tutorial\tutorial\Debug\tutorial.sni

Notes: 1. This section describes general usage examples of the emulator (H8/3048F-ONE E10T emulator). For each product specifications, refer to section 6 or on-line help.

2. This program was created by using Hitachi Embedded Workshop (hereafter referred to as HEW) V1.2. Older versions of HEW will not open the workspace included with the package, so create a new workspace in such situations.

3. This program was compiled without optimization. If recompiled with different settings, the addresses may differ from those given in this section.

4. `tutorial.abs` is a load module in the Dwarf2 format. If a load module is recreated in the Sysrof format, the amount of information displayed on the HDI screen during the program's execution will be reduced.

5. This section describes general usage examples for the emulator. For the specifications of particular products, refer to section 6 or the online help file.

3.2 Running the HDI

To run the HDI, select [HDI for E10T H8xxxx] -> [Hitachi Debugging Interface] from the [Start] menu.

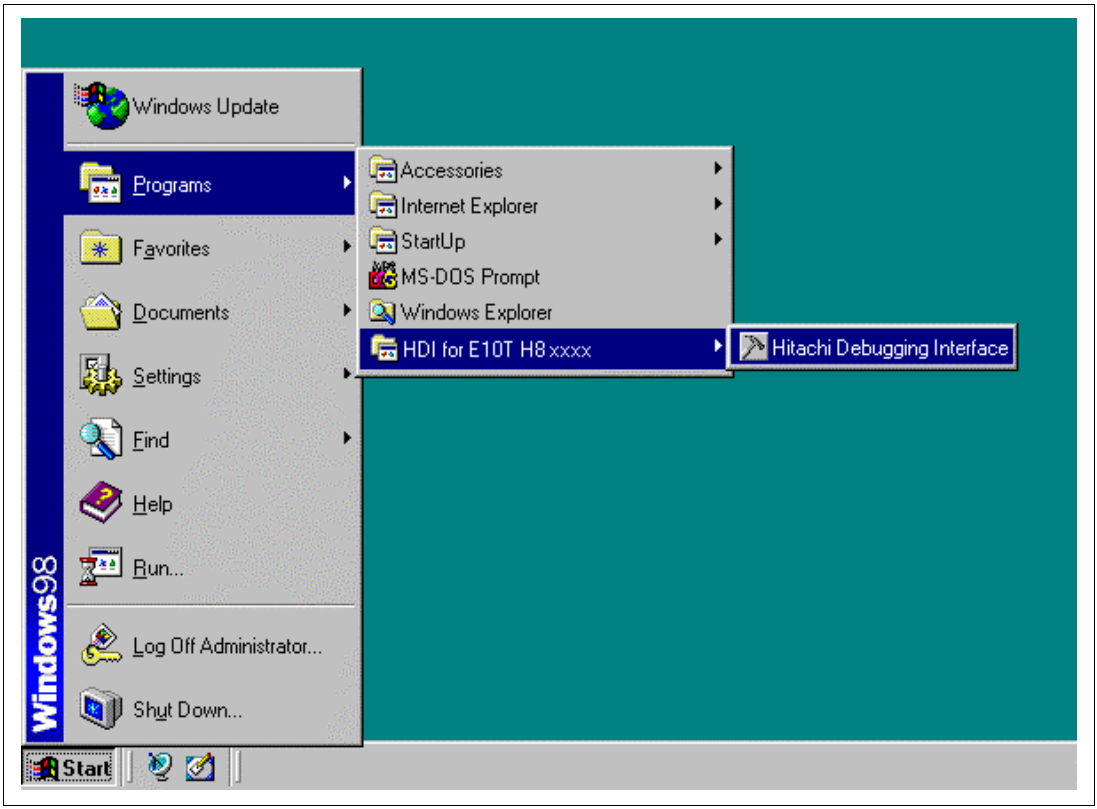


Figure 3.1 [Start] Menu

For the procedure of running the HDI, refer to section 2.5, System Check.

3.3 [HDI] Window

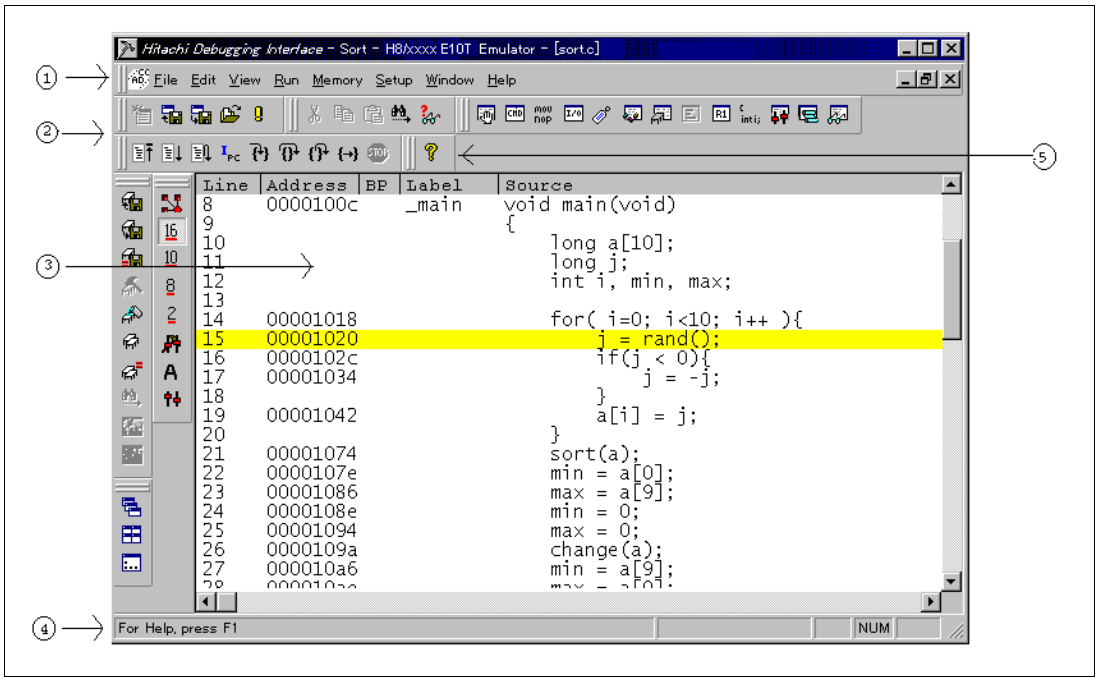


Figure 3.2 [HDI] Window

The key functions of the HDI are described in section 4, Descriptions of Windows. Numbers in figure 3.2 indicate the following:

1. Menu bar: Gives the user access to the HDI commands for using the HDI debugger.

2. Toolbar: Provides convenient buttons as shortcuts for the most frequently used menu commands.

3. Program window: Displays the source program being debugged.

4. Status bar: Displays the status of the emulator, and progress information about downloading.

5. [Help] button: Activates on-line help about any features of the HDI user interface.

3.4 Setting up the Emulator

The following MCU conditions must be set up on the emulator before downloading the program:

- Device type
- Execution mode

The following describes how to set up the emulator for the tutorial programs.

3.5 Setting the [Configuration] Dialog Box

- Select [Configure Platform...] from the [Setup] menu to set configuration. The [Configuration] dialog box is displayed.

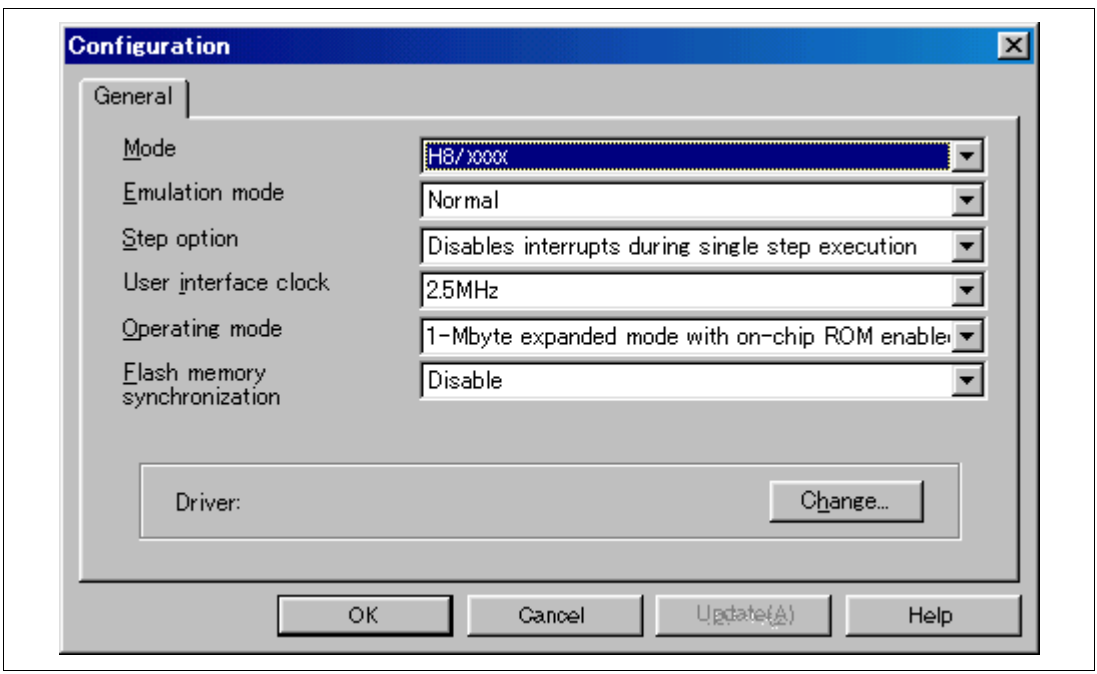


Figure 3.3 [Configuration] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Set options as follows:

Table 3.2 Setting the [Configuration] Dialog Box

Option	Value
Mode	H8/xxxx (default)
Emulation mode	Normal (normal execution, default)
Step option (Disables interrupt during single step execution)	Disables interrupts during single step execution (default)
User interface clock	According to the system clock frequency, the maximum user interface clock is automatically set.
Operating mode	Selected according to the settings of the MD pin.
Break condition mode	Emulator (default)
Flash memory synchronization	Disable (default)

- Click the [OK] button to set any changes in the configuration.

3.6 Downloading the Tutorial Program

3.6.1 Downloading the Tutorial Program

Download the object program to be debugged.

- Select [Load Program...] from the [File] menu. The [Load Program] dialog box is displayed. Enter the [Offset] edit box and [File name] list box as shown in figure 3.4 and click the [Open] button.

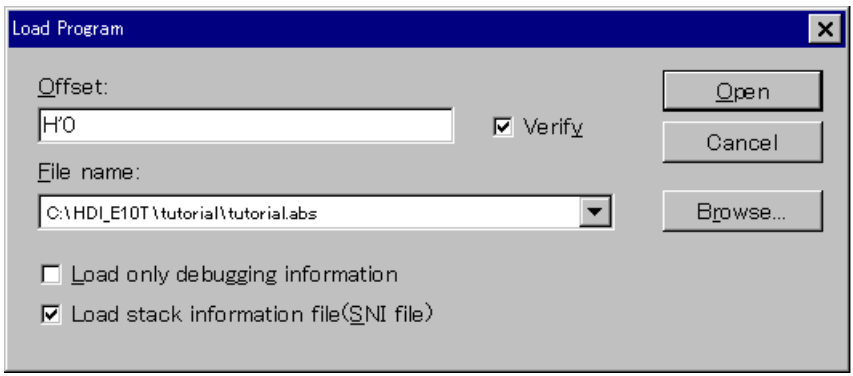


Figure 3.4 [Load Program] Dialog Box

When the file has been loaded, the following dialog box displays information about the memory areas that have been filled with the program code.

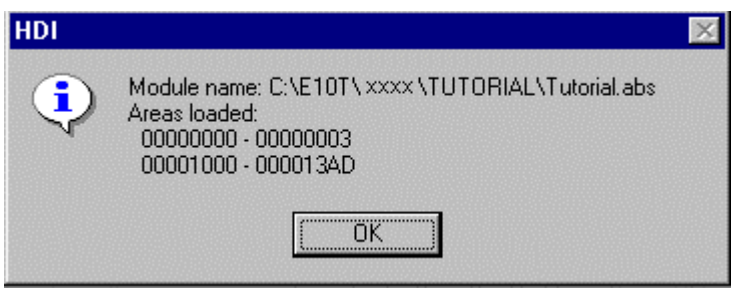


Figure 3.5 [HDI] Dialog Box

- Click the [OK] button to continue.

3.6.2 Displaying the Source Program

The HDI allows the user to debug a program at the source level.

- Select [Source...] from the [View] menu. The [Open] dialog box is displayed.
- Select the C source file that corresponds to the object file the user has loaded.

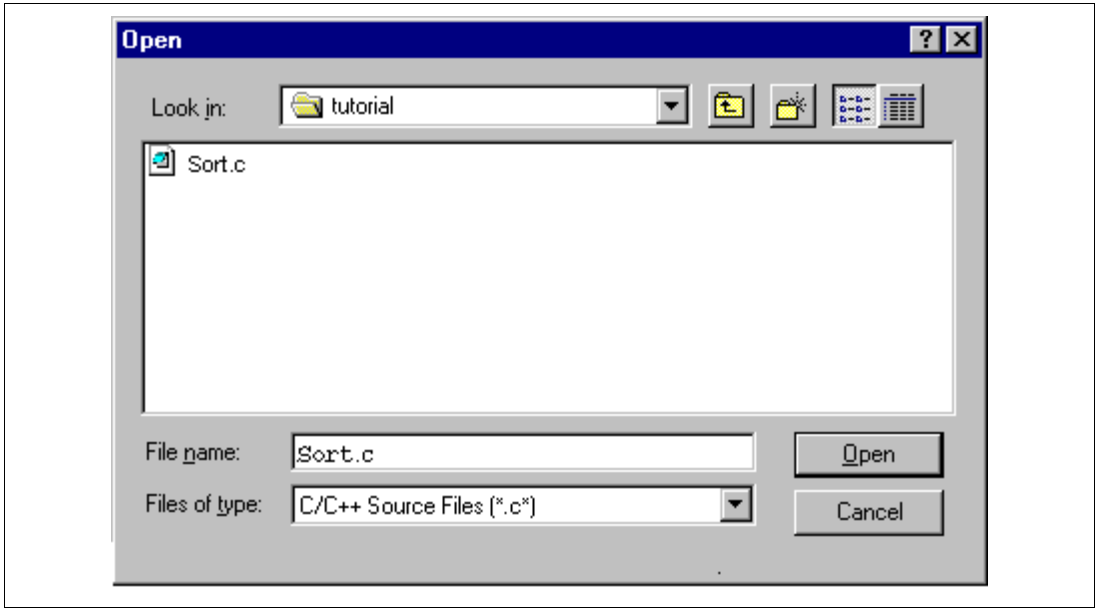


Figure 3.6 [Open] Dialog Box

- Select [Sort.c] and click the [Open] button. The [Program] window is displayed.

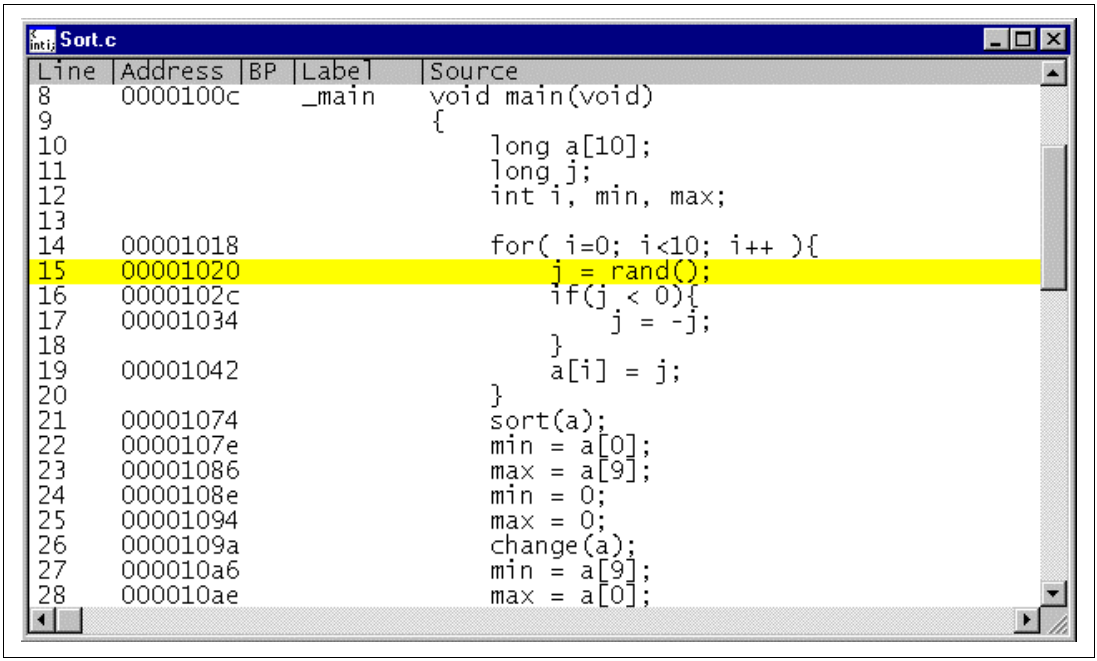


Figure 3.7 [Program] Window (Displaying the Source Program)

- If necessary, select the [Font] option from the [Customize] submenu on the [Setup] menu to select a clear font and size.

Initially the [Program] window shows the start of the main program, but the user can use the scroll bar to scroll through the program to see the other statements.

3.7 Setting the Software Breakpoint

A breakpoint is one of the easy debugging functions.

The [Program] window provides a very simple way of setting a software breakpoint at any point in a program. For example, to set a breakpoint at the `sort` function call:

- Select by double-clicking the [BP] column on the line containing the `sort` function call.

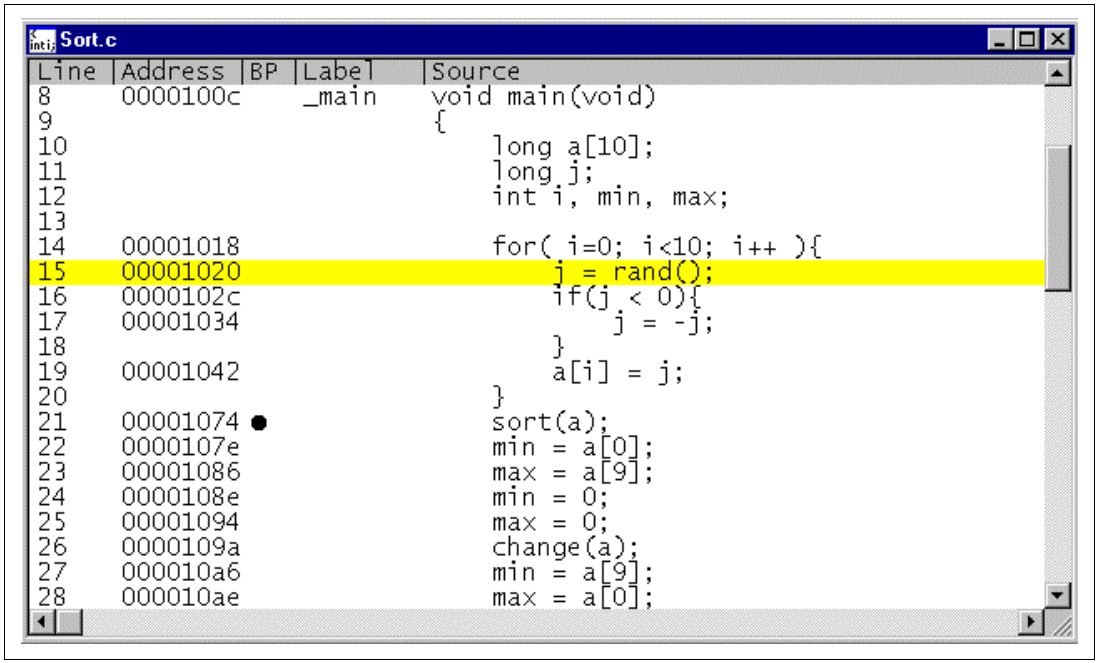


Figure 3.8 [Program] Window (Setting a Software Breakpoint)

The word [● `Break`] will be displayed on the line containing the `sort` function to show that a software breakpoint is set.

3.8 Setting Registers

Set values of the program counter and the stack pointer before executing the program.

- Select [Registers] from the [View] menu. The [Registers] window is displayed.

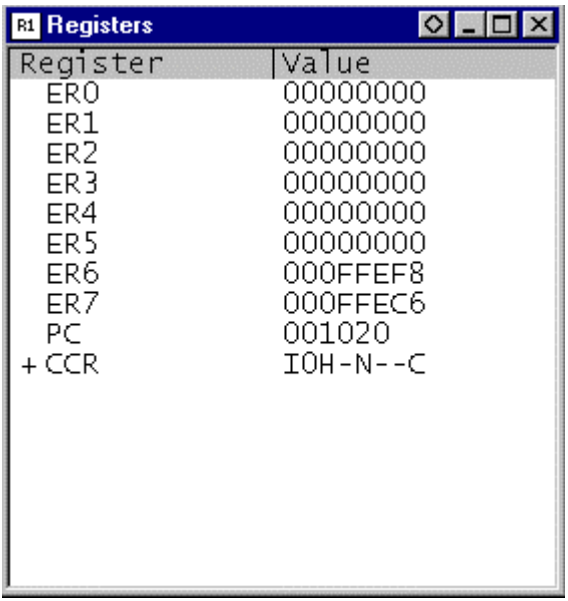


Figure 3.9 [Registers] Window

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

- To change the value of the program counter (PC), double-click the value area in the [Registers] window with the mouse. The following dialog box is then displayed, and the value can be changed.

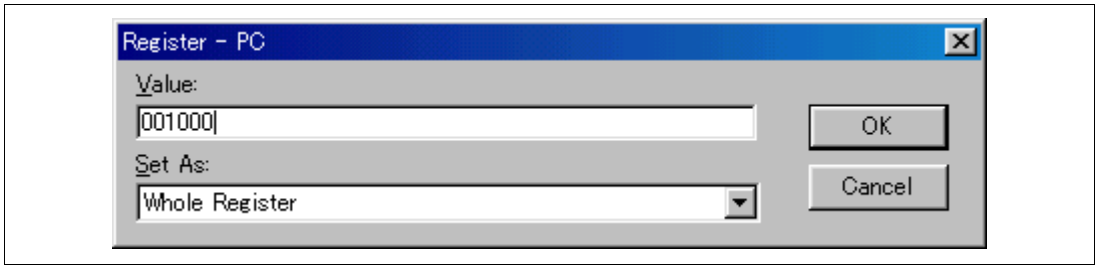


Figure 3.10 [Register] Dialog Box (PC)

- Set the program counter to H'1000 in this tutorial program, and click the [OK] button.
- To change the value of the stack pointer (SP), move the mouse pointer on the value to be changed in the [ER7] value area in the [Registers] window and enter the new value by the keyboard, or double-click the value area with the mouse. The following dialog box is then displayed.

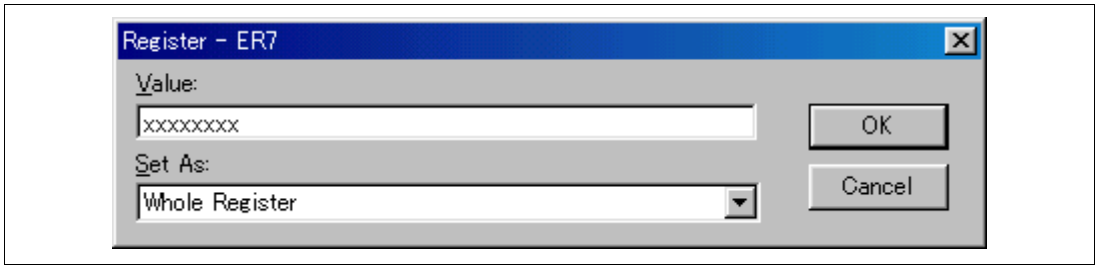


Figure 3.11 [Register] Dialog Box (ER7)

- Set the value so that the stack area is allocated on the RAM, and click the [OK] button.

3.9 Executing the Program

Execute the program as described in the following:

- To execute the program, select [Go] from the [Run] menu, or click the [Go] button on the toolbar.

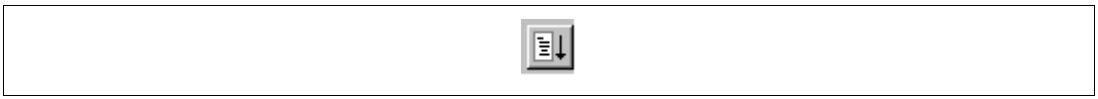


Figure 3.12 [Go] Button

- Or, select [Reset Go] from the [Run] menu or click the [Reset Go] button on the toolbar.

Figure 3.13 [Reset Go] Button

The program will be executed up to the breakpoint that has been inserted, and a statement will be highlighted in the [Program] window to show the position that the program has halted, with the message [Break=BREAKPOINT] in the status bar.

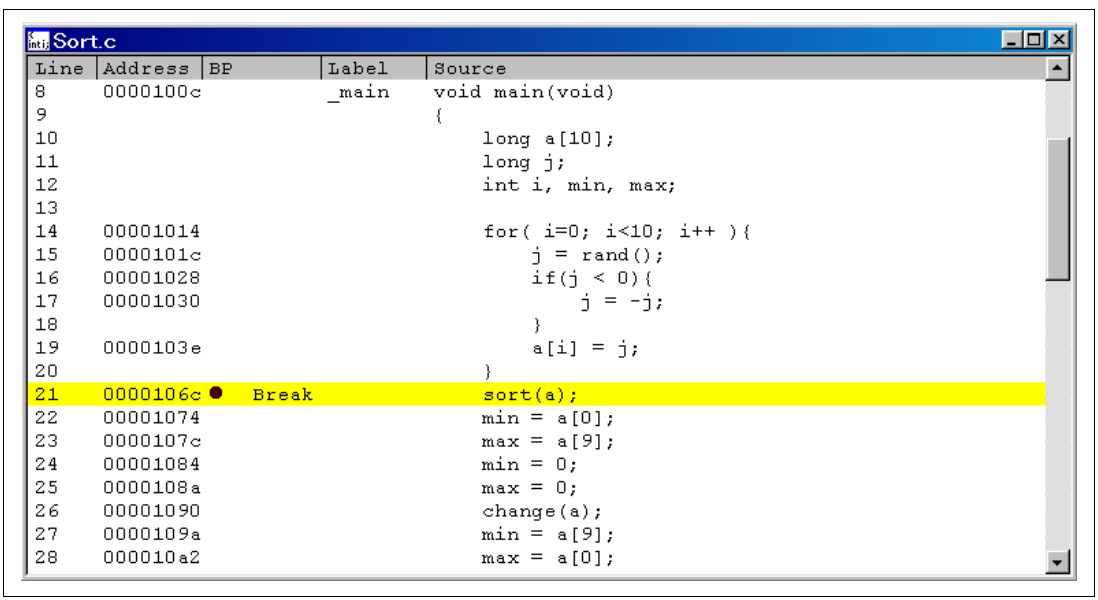


Figure 3.14 [Program] Window (Break Status)

The user can see the cause of the break that occurred last time in the [System Status] window.

- Select [Status] from the [View] menu. After the [System Status] window is displayed, open the [Platform] page, and check the status of Cause of last break.

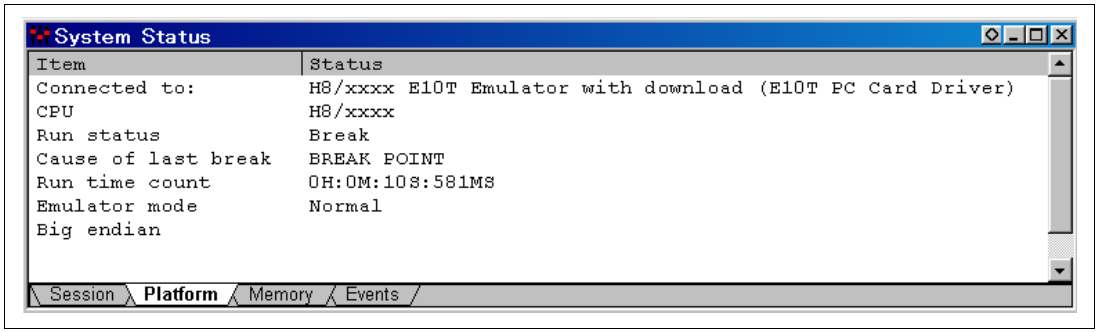


Figure 3.15 [System Status] Window

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the on-line help.

The [System Status] window displays the following items in each page.

Table 3.3 Contents of the [System Status] Window

Page	Item	Description
[Session]	Target System	Always displays Connected.
	Session Name	Displays the session file name.
	Program Name	Displays the load module file name.
[Platform]	Connected To:	Displays the name of the connected emulator and the selected driver name.
	CPU	Displays the target MCU name.
	Run status	Displays the execution status: RUNNING: Being executed Break: Stopped
	Cause of last break	Displays the cause of the emulator stopping at break. In this example, the cause of the stop is BREAK POINT.
	Run time count	Displays the program execution time. The display format is H: hours, M: minutes, S: seconds, and MS: milliseconds. In this example, 0H:0M:10S:581MS is displayed.
	Emulator mode	Displays the emulator operating mode (setting information for [Emulation Mode] of the [Configuration] dialog box).
[Memory]	Loaded Memory Areas	Displays the loaded area of the load module.
[Events]	Resources	Displays the usage states of BREAKPOINT and Break Condition.

3.10 Reviewing Breakpoints

The user can see all the breakpoints set in the program in the [Breakpoints] window.

- Select [Breakpoints] from the [View] menu.

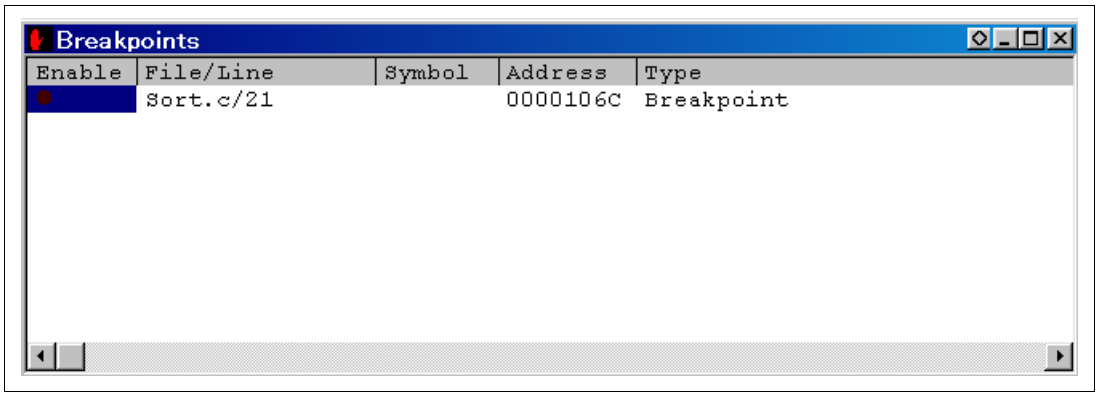


Figure 3.16 [Breakpoints] Window

The pop-up menu, opened by clicking the [Breakpoints] window with the right mouse button, also allows the user to set or change breakpoints, define new breakpoints, and delete, enable, or disable breakpoints.

3.11 Viewing Memory

The user can view the contents of a memory block in the [Memory] window. For example, to view the memory contents corresponding to the `main` in word size:

- Select [Memory …] from the [View] menu, enter ***`main`*** in the [Address] edit box, and set `Word` in the [Format] combo box.

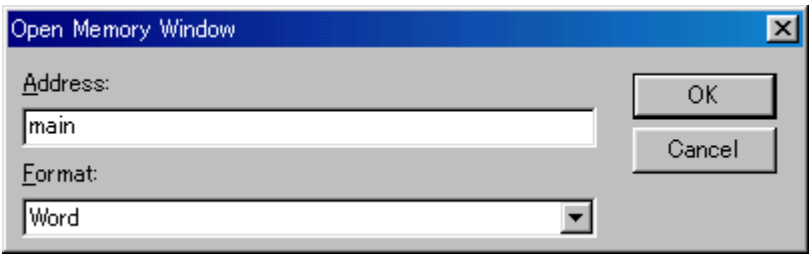


Figure 3.17 [Open Memory Window] Dialog Box

- Click the [OK] button. The [Memory] window showing the specified area of memory is displayed.

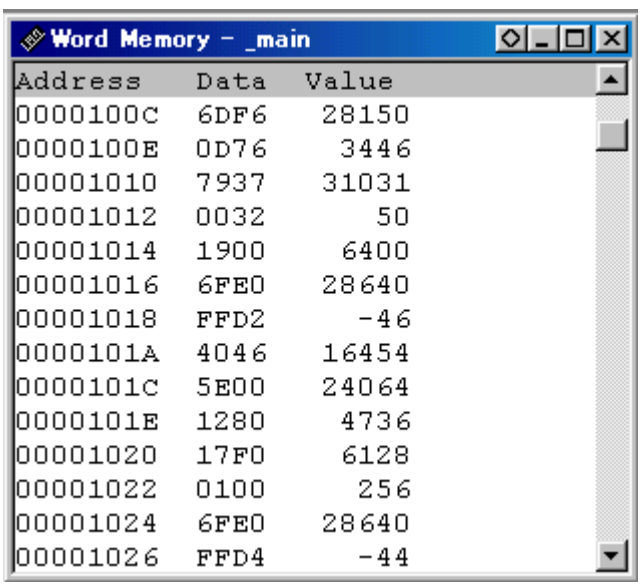


Figure 3.18 [Memory] Window

3.12 Watching Variables

As the user steps through a program, it is possible to watch that the values of variables used in the user program are changed. For example, set a watch on the long-type array a declared at the beginning of the program, by using the following procedure:

- Click the left of displayed array a in the [Program] window to position the cursor.
- Click the [Program] window with the right mouse button and select [Instant Watch...] from a pop-up menu.

  The following dialog box will be displayed.

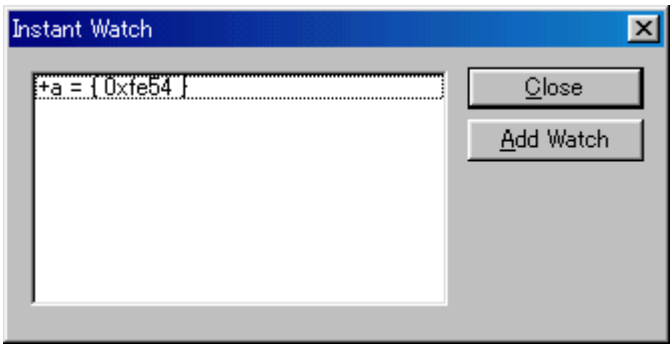


Figure 3.19 [Instant Watch] Dialog Box

- Click [Add Watch] button to add a variable to the [Watch] window.

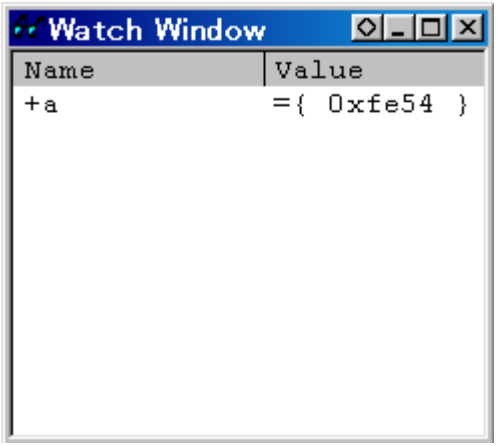


Figure 3.20 [Watch] Window (Displaying the Array)

The user can also add a variable to the [Watch] window by specifying its name.

- Click the [Watch] window with the right mouse button and select [Add Watch] from the pop-up menu.

  The following dialog box will be displayed.

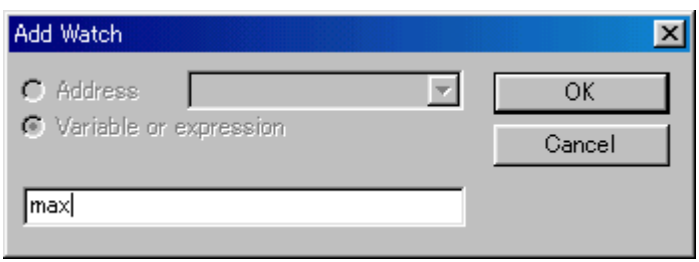


Figure 3.21 [Add Watch] Dialog Box

- Input variable ***max*** and click the [OK] button.

The [Watch] window will now also show the int-type variable max.

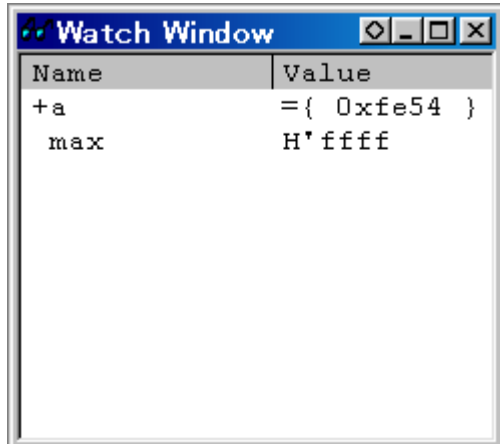


Figure 3.22 [Watch] Window (Displaying the Variable)

The user can double-click the + symbol to the left of any variable in the [Watch] window to watch the all elements in array a.

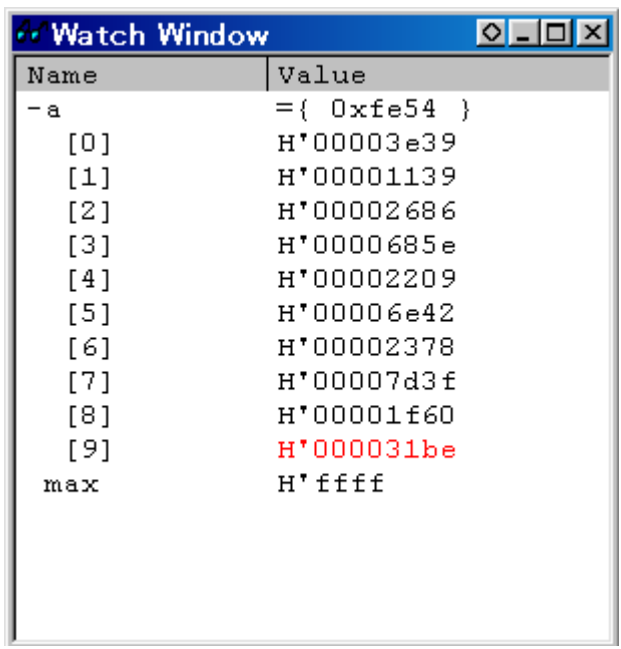


Figure 3.23 [Watch] Window (Displaying Array Elements)

3.13 Stepping Through a Program

The HDI provides a range of step menu commands that allow efficient program debugging.

Table 3.4 Step Option

Menu Command	Description
Step In	Executes each statement, including statements within functions.
Step Over	Executes a function call in a single step.
Step Out	Steps out of a function, and stops at the statement following the statement in the program that called the function.
Step…	Steps the specified times repeatedly at a specified rate.

3.13.1 Executing [Step In] Command

The [Step In] steps into the called function and stops at the first statement of the called function.

- To step through the `sort` function, select [Step In] from the [Run] menu, or click the [Step In] button in the toolbar.

Figure 3.24 [Step In] Button

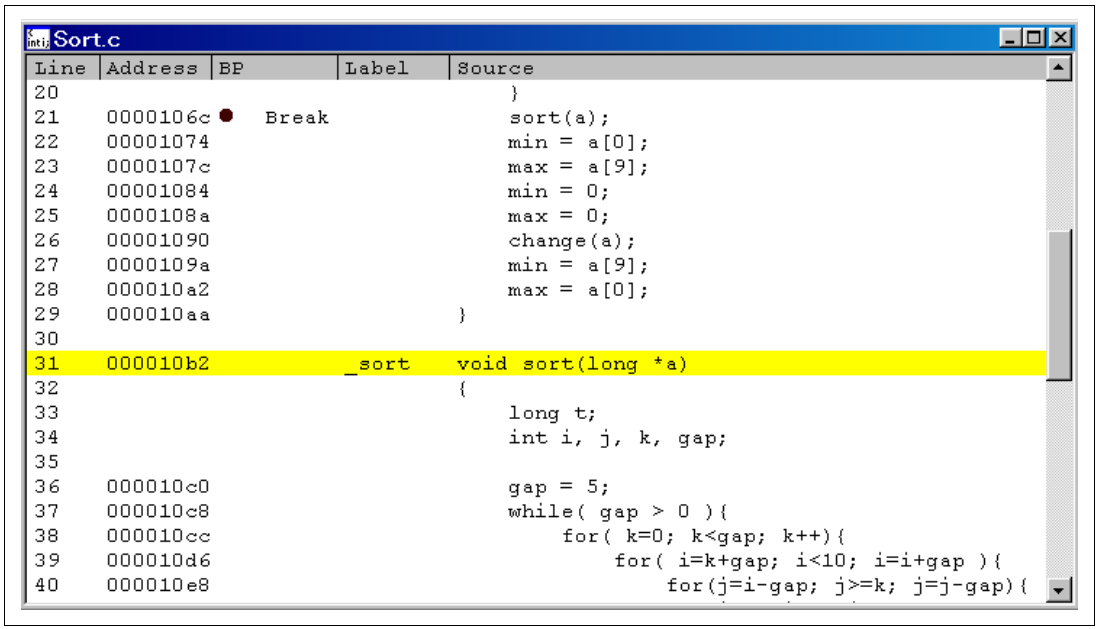


Figure 3.25 [Program] Window (Step In)

- The highlighted line moves to the first statement of the `sort` function in the [Program] window.

3.13.2 Executing [Step Out] Command

The [Step Out] steps out of the called function and stops at the next statement of the calling statement in the main function.

- To step out of the `sort` function, select [Step Out] from the [Run] menu, or click the [Step Out] button in the toolbar.

Figure 3.26 [Step Out] Button

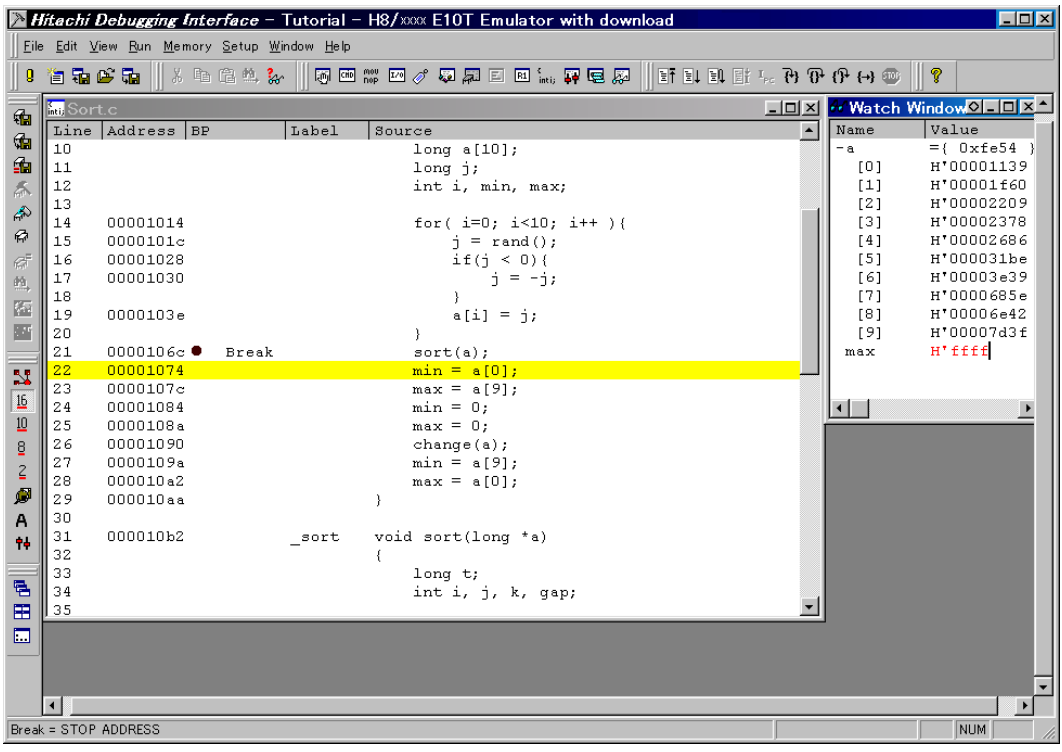


Figure 3.27 [Program] Window (Step Out)

- The data of variable `a` displayed in the [Watch] window is sorted in ascending order.

- To execute two steps, use [Step In] twice.

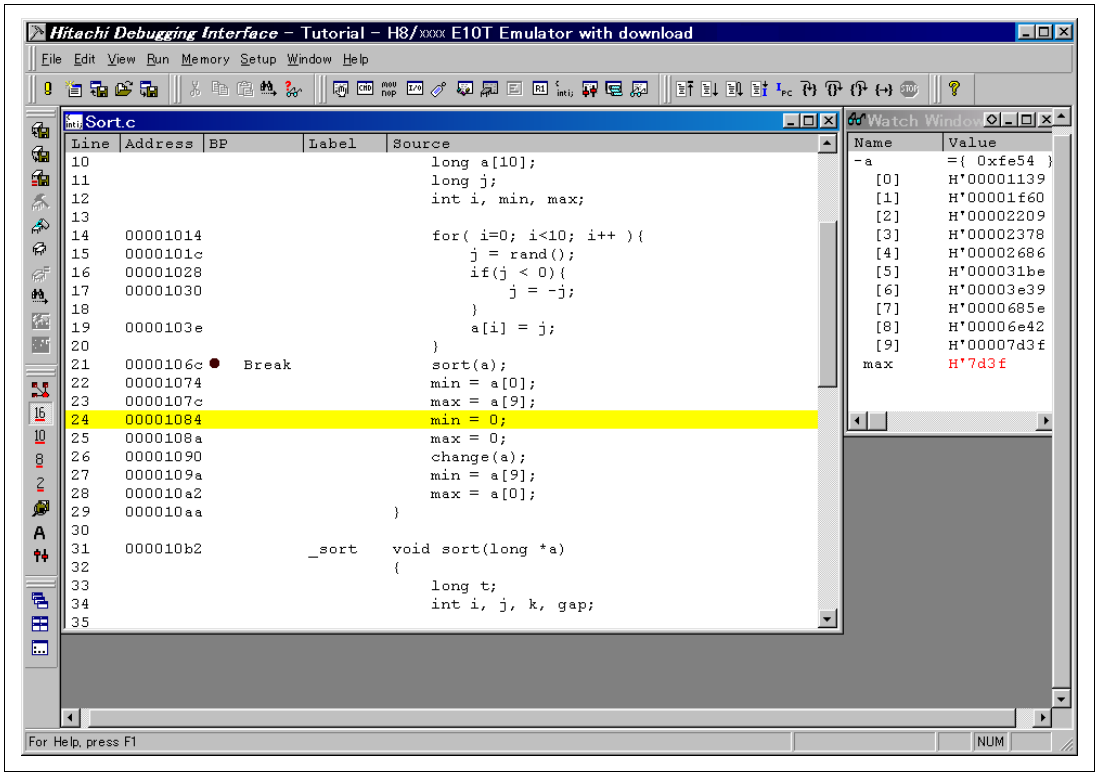


Figure 3.28 [Program] Window (Step In –> Step In)

- The value of `max` displayed in the [Watch] window is changed to the maximum data value.

3.13.3 Executing [Step Over] Command

The [Step Over] executes a function call as a single step and stops at the next statement of the main program.

- Using [Step Over], execute two steps to reach the change function statement.

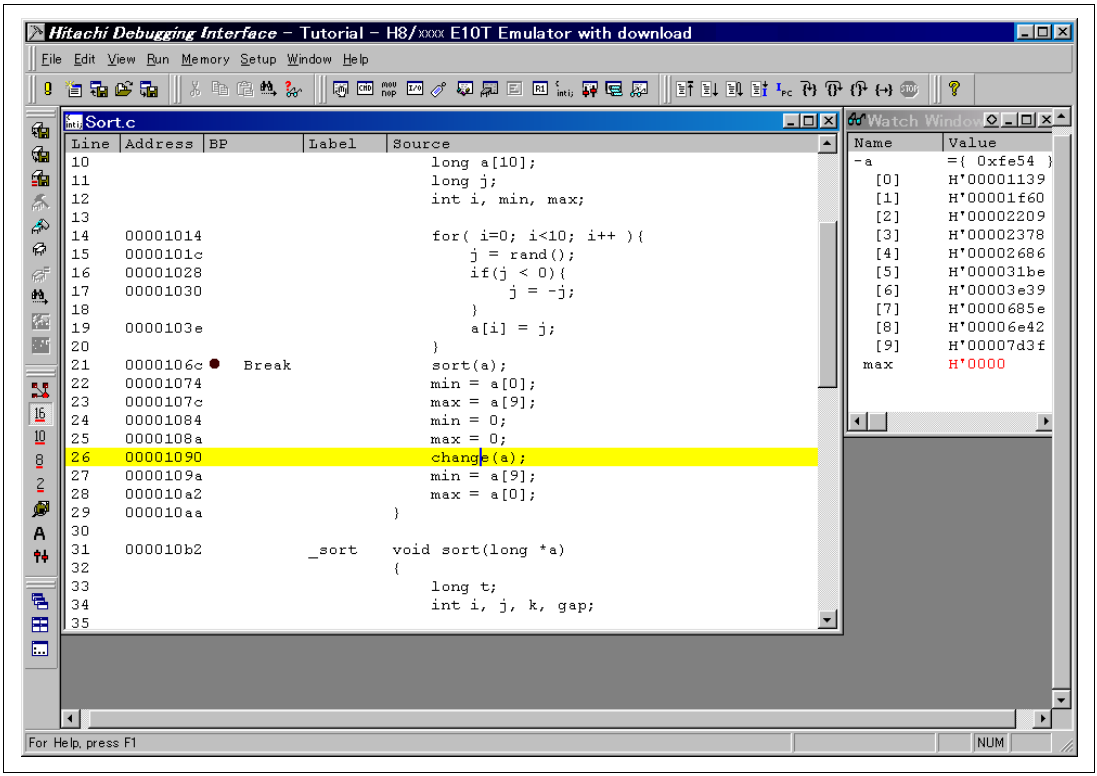


Figure 3.29 [Program] Window (Before Step Over Execution)

- To step through all statements in the change function at a single step, select [Step Over] from the [Run] menu, or click the [Step Over] button in the toolbar.

Figure 3.30 [Step Over] Button

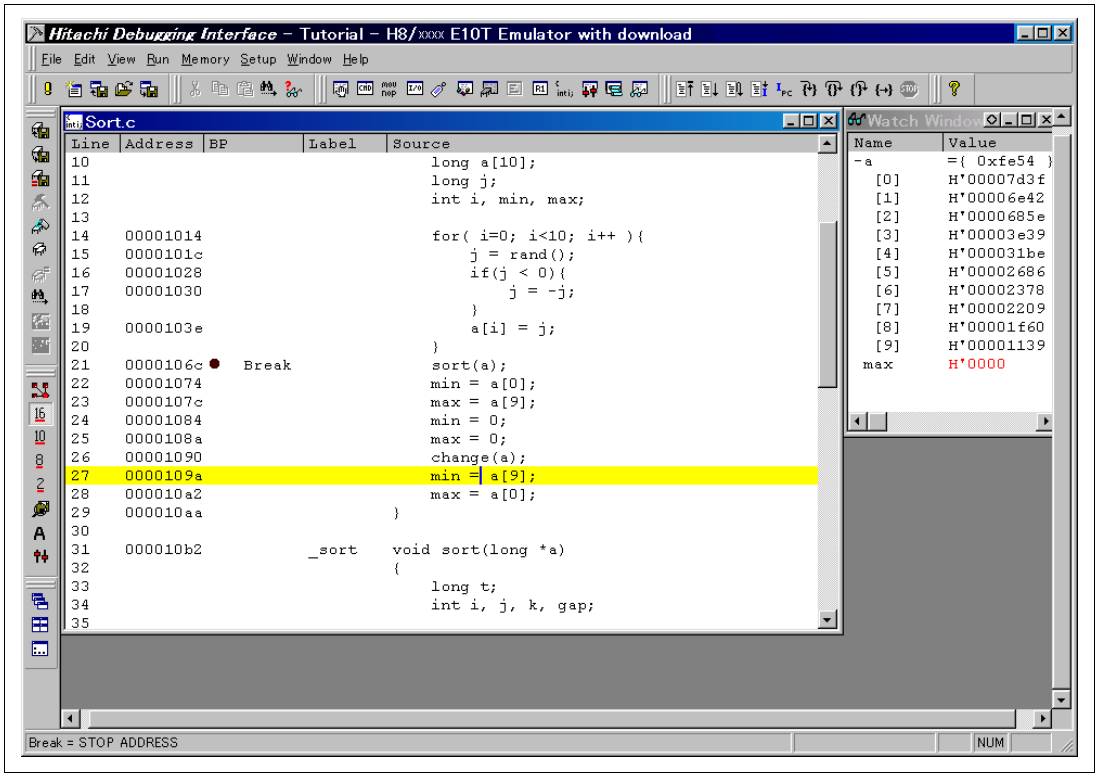


Figure 3.31 [Program] Window (Step Over)

When the last statement of the change function is executed, the data of variable a, which is displayed in the [Watch] window, is sorted in descending order.

3.14 Displaying Local Variables

The user can display local variables in a function using the [Locals] window. For example, we will examine the local variables in the `main` function, which declares five local variables: `a`, `j`, `i`, `min`, and `max`.

- Select [Locals] from the [View] menu. The [Locals] window is displayed.

  Initially, the [Locals] window is empty because local variables have not yet been declared.

- Select [Step In] from the [Run] menu to execute a single step.

  The [Locals] window will now show the local variables and their values.

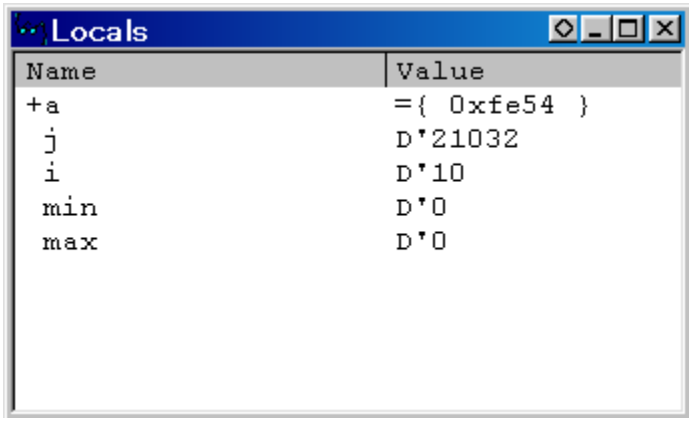


Figure 3.32 [Locals] Window

- Double-click the + symbol in front of array `a` in the [Locals] window to display the elements of array `a`.
- Refer to the elements of array `a` before and after the execution of the `sort` function, and confirm that random data is sorted in descending order.

3.15 Break Function

The emulator has software and hardware break functions. With the HDI, a software breakpoint can be set using the [Breakpoints] window, and a hardware break condition can be set using the [Break condition 1] dialog box.

An overview and setting of the break function are described below.

3.15.1 Software Break Function

The emulator can set up to 255 software breakpoints. Setting a software breakpoint is described below.

- Select [Breakpoints] from the [View] menu. The [Breakpoints] window is displayed.
- Click the [Breakpoints] window with the right mouse button and select [Delete All] from the pop-up menu to cancel all the breakpoints that have been set.

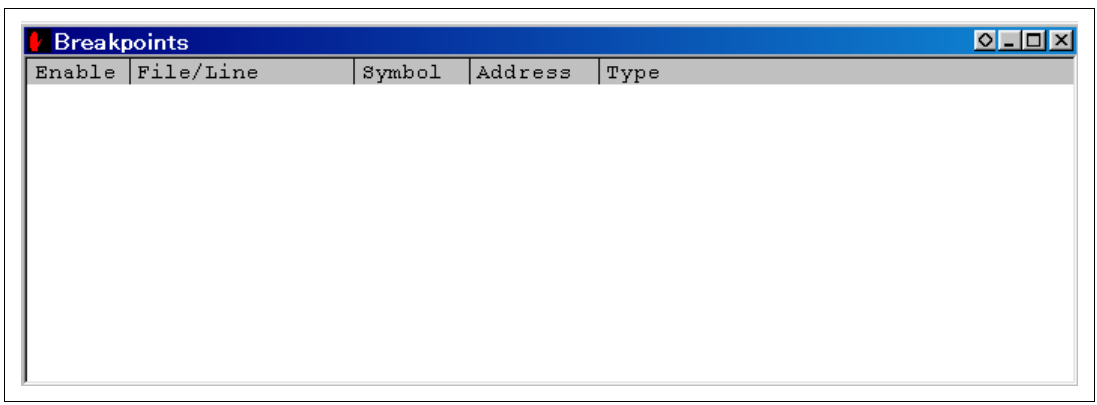


Figure 3.33 [Breakpoints] Window (Before Software Breakpoint Setting)

- Click the [Breakpoints] window with the right mouse button and select [Add] from the pop-up menu.

The [Break] dialog box is displayed. The [Point] page is displayed as a default.

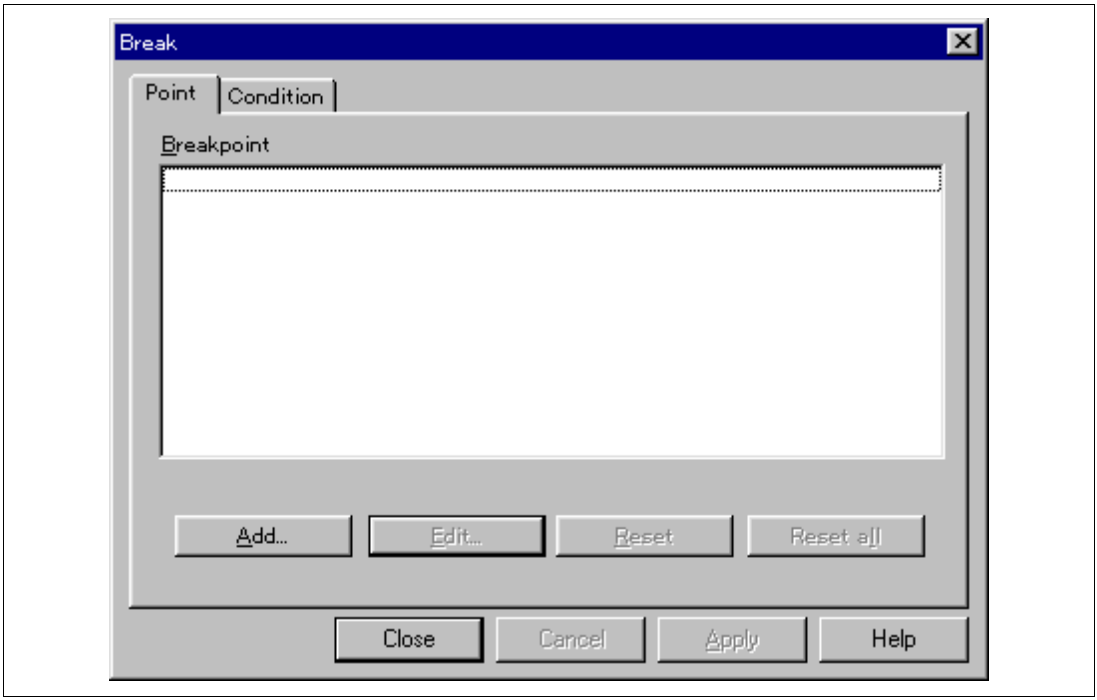


Figure 3.34 [Point] Page ([Break] Dialog Box)

- Click the [Add...] button to display the [Breakpoint] dialog box.
- Enter ***H'109a*** to the [Value] edit box.

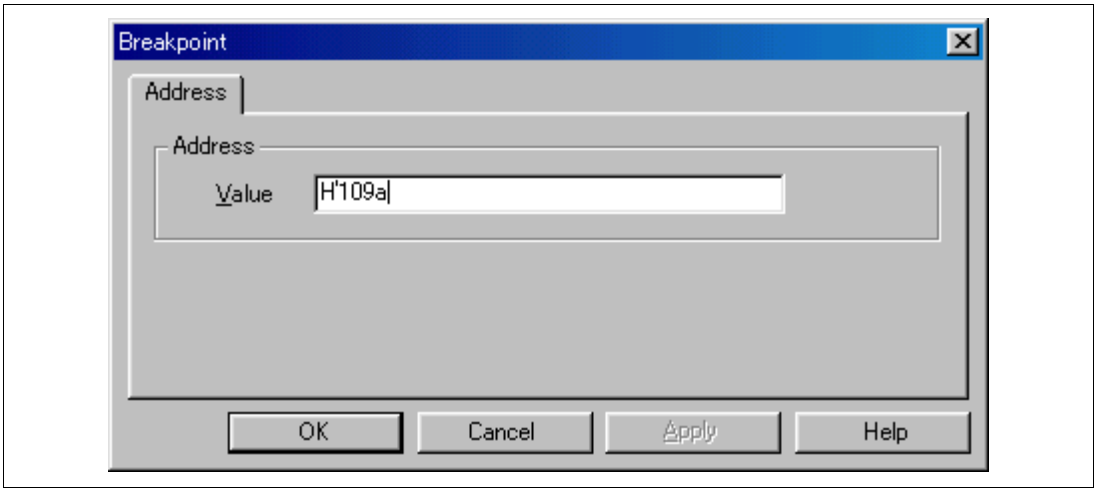


Figure 3.35 [Breakpoint] Dialog Box

- Click the [OK] button.

The [Break] dialog box is displayed. The address set in the value field of [Breakpoint] is displayed.

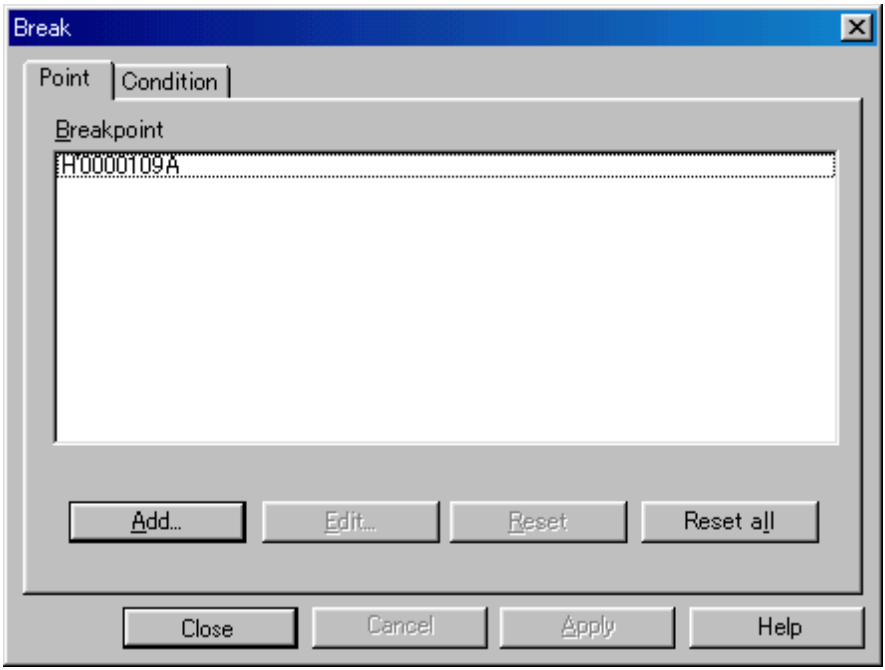


Figure 3.36 [Point] Page ([Break] Dialog Box) (After Software Breakpoint Setting)

- Click the [Close] button.

The software breakpoint that has been set is displayed in the [Breakpoints] window.

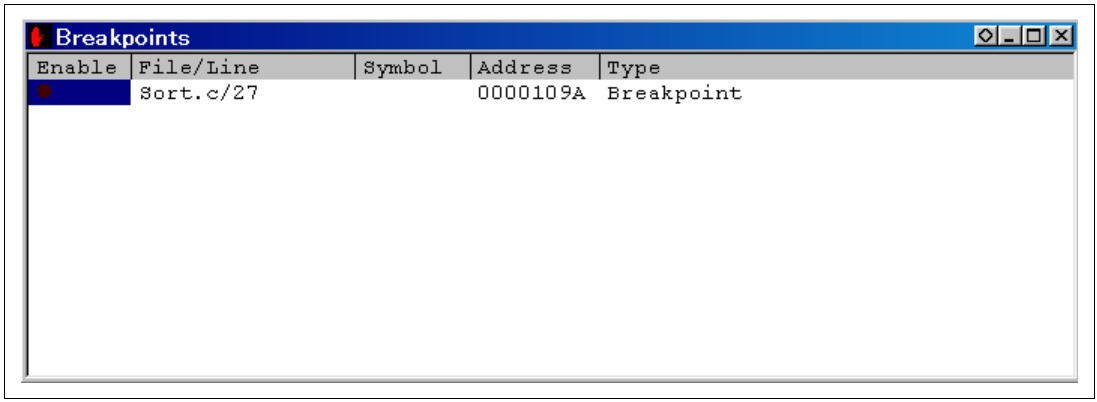


Figure 3.37 [Breakpoints] Window (Software Breakpoint Setting)

To stop the tutorial program at the breakpoint, the following procedure must be executed:

- Close the [Breakpoints] window.
- Click the [Reset Go] button.

The program runs, and stops at the set breakpoint.

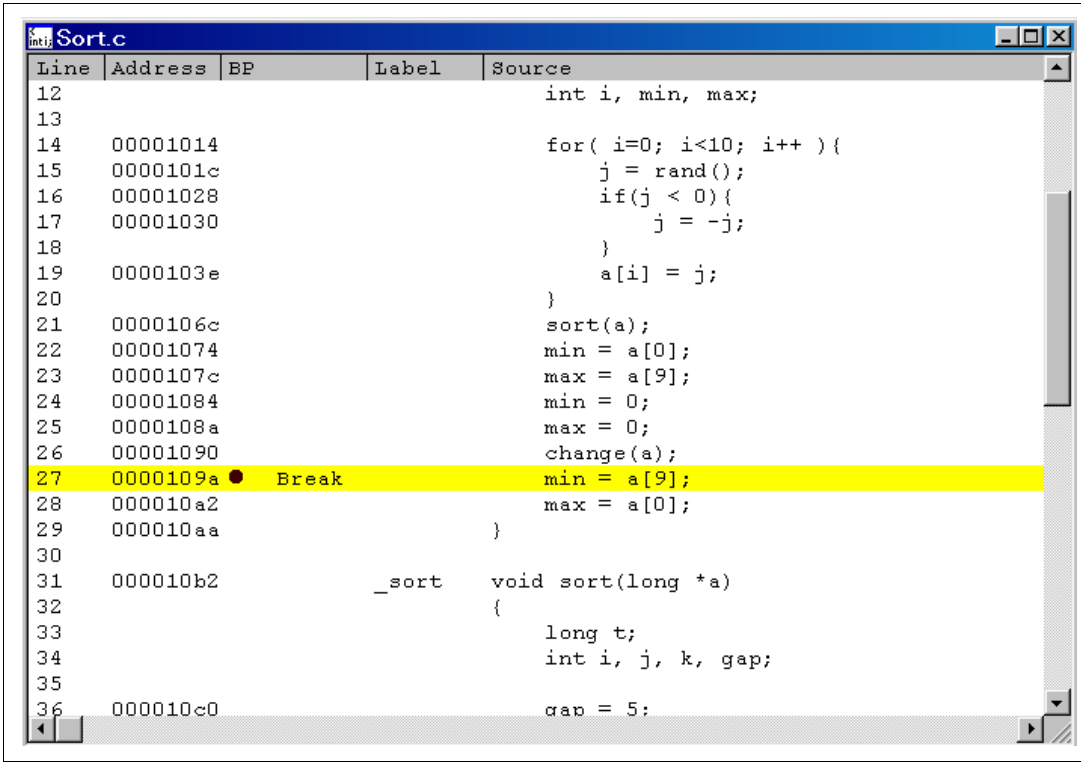


Figure 3.38 [Program] Window at Execution Stop (Software Break)

The [System Status] window displays the following contents.

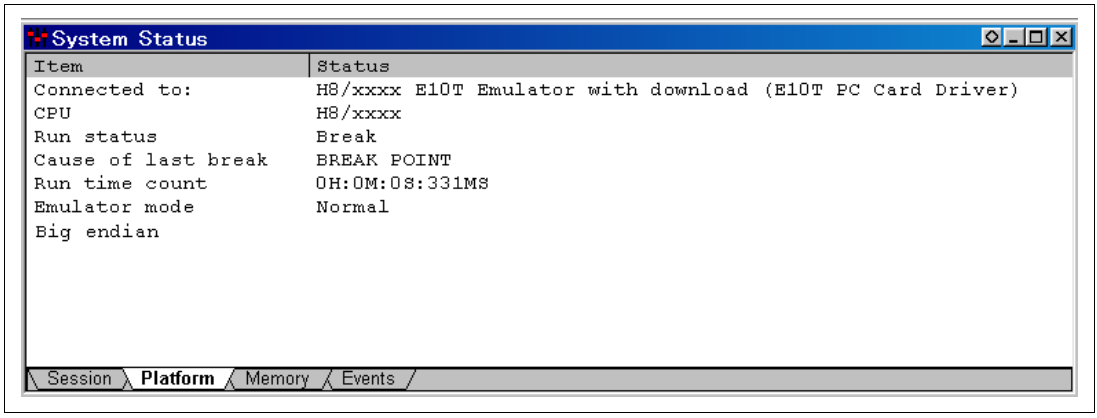


Figure 3.39 Displayed Contents of the [System Status] Window (Software Break)

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the on-line help.

3.16 Hardware Break Function

A method is given below in which the address bus condition and the read cycles for the state condition are set under Break condition 1 as hardware break conditions.

- Select [Breakpoint Window] from the [View] menu. The [Breakpoints] window is displayed.
- Click the [Breakpoints] window with the right mouse button and select [Delete All] from the pop-up menu to cancel all breakpoints that have been set.
- Click the [Breakpoints] window with the right mouse button and select [Add] from the pop-up menu.

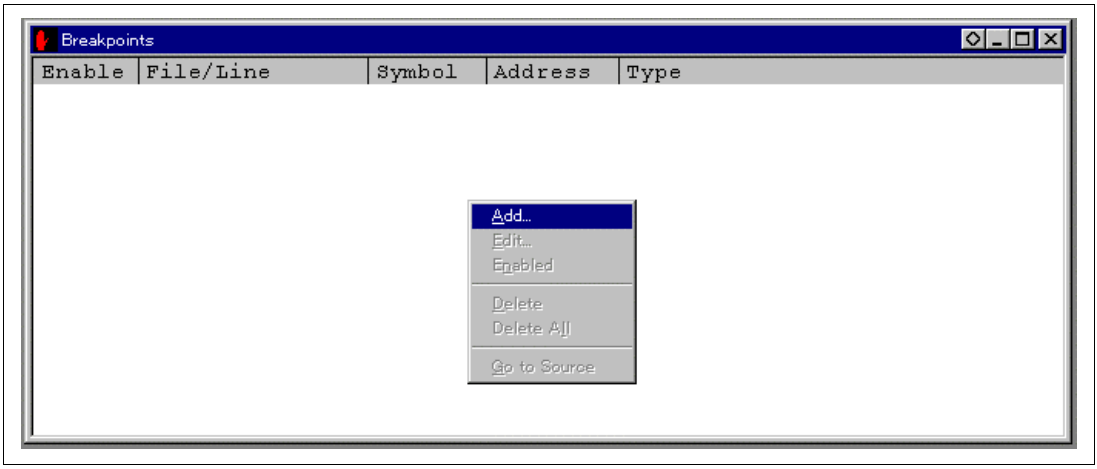


Figure 3.40 [Breakpoints] Window (Before Hardware Break Condition Setting)

The [Break] dialog box is displayed. To set hardware break conditions, select [Condition] in the [Break] dialog box to display the [Condition] page.

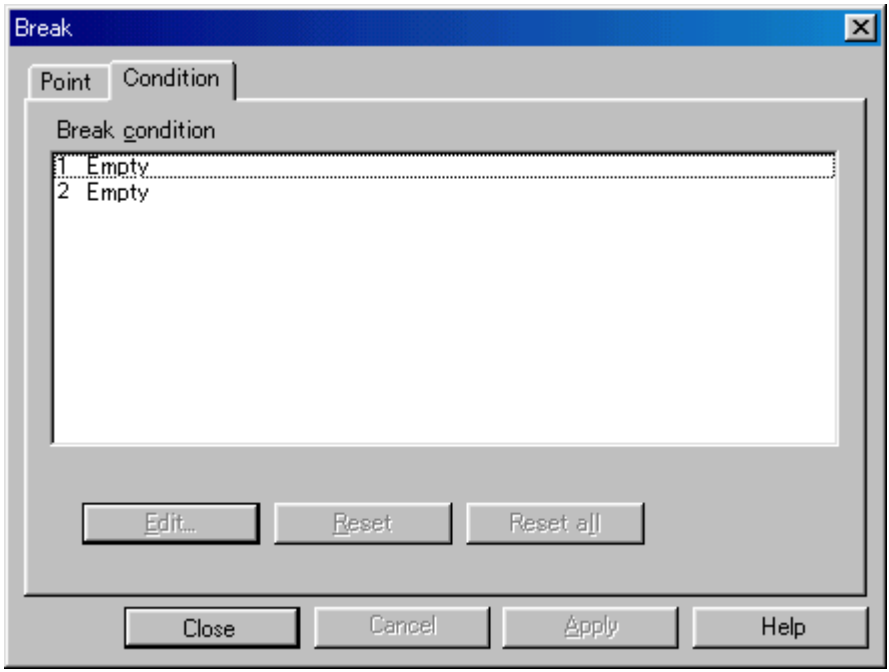


Figure 3.41 [Condition] Page ([Break] Dialog Box)

Up to two breakpoints can be set independently for the hardware break condition. In this example, set the hardware break condition for Break condition 1.

Note: Note that the number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the on-line help.

- Highlight the first point in the [Break condition] list box.
- Click the [Edit...] button. The [Break condition 1] dialog box is displayed.
- Clear the [Don't care] check box in the [Address] page.
- Select the [Only program fetched address after] radio button and enter ***H'1084*** as the value in the [Address] edit box.
- Select the [Read] radio button in the [Read/Write] group box.

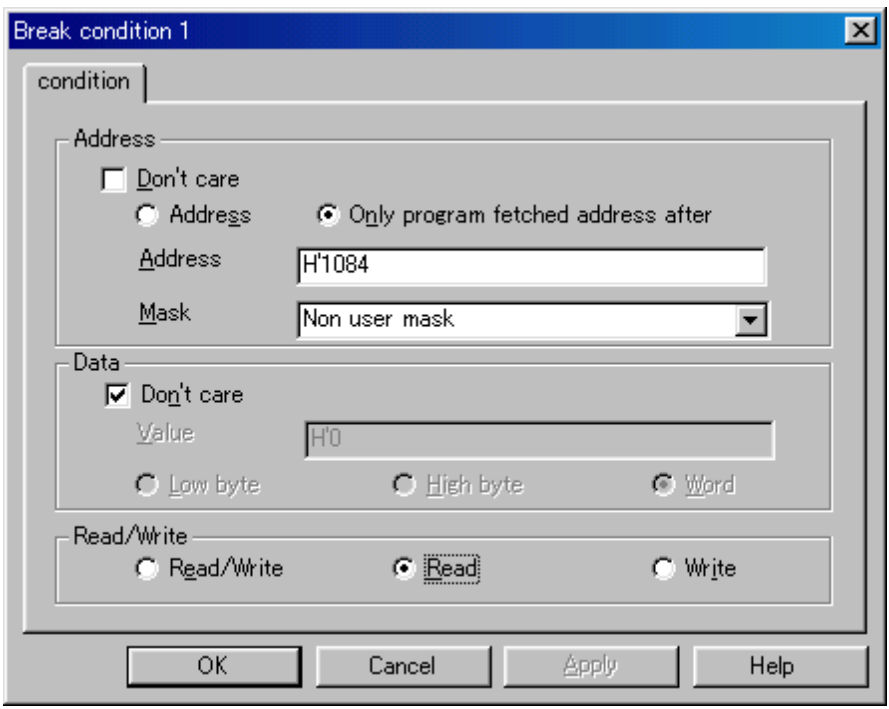


Figure 3.42 [condition] Page ([Break condition 1] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

- Click the [OK] button.
- The [Break] dialog box is displayed. Check the first point display in the [Break condition] list box is changed from `Empty` to `Enable`.

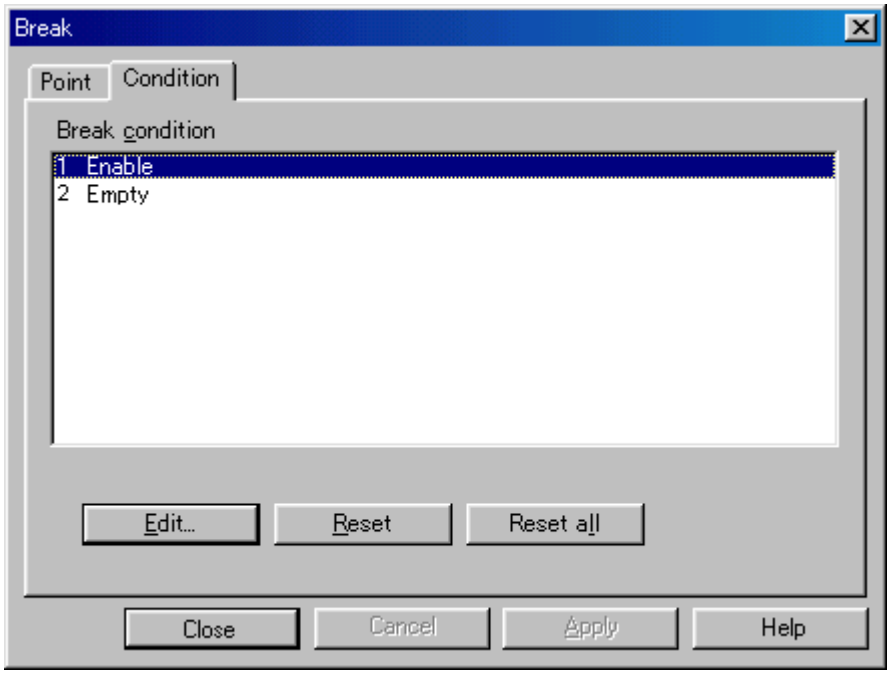


Figure 3.43 [Break] Dialog Box (After Hardware Break Condition Setting)

Note: Note that the number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the on-line help.

- Click the [Close] button.

The newly set hardware breakpoint is displayed in the [Breakpoints] window. With this setting, `Break condition 1` is displayed in [Type] in the [Breakpoints] window.

This completes the setting of the Break condition 1 hardware break condition. When the program is executed, a break will occur when address H'1084 is accessed in a read cycle.

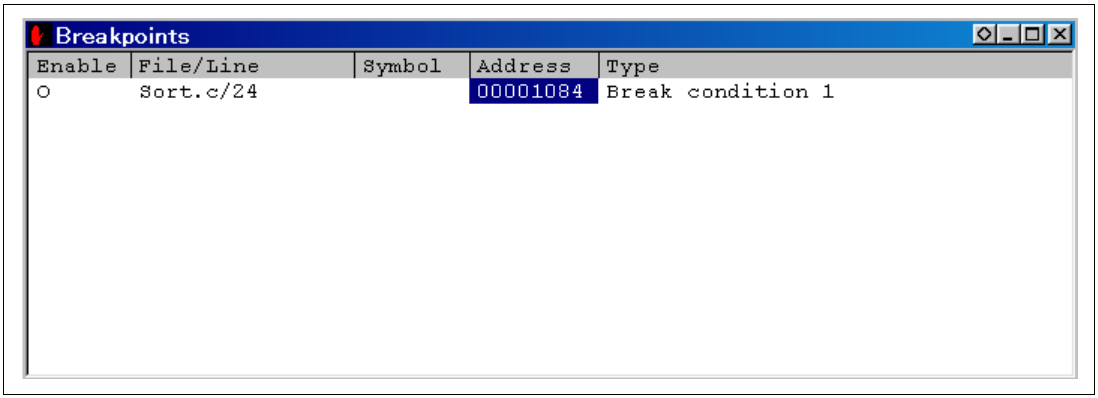


Figure 3.44 [Breakpoints] Window ([Break condition 1] Setting)

- Close the [Breakpoints] window.
- Click the [Reset Go] button.

The program runs then stops at the condition specified under Break condition 1.

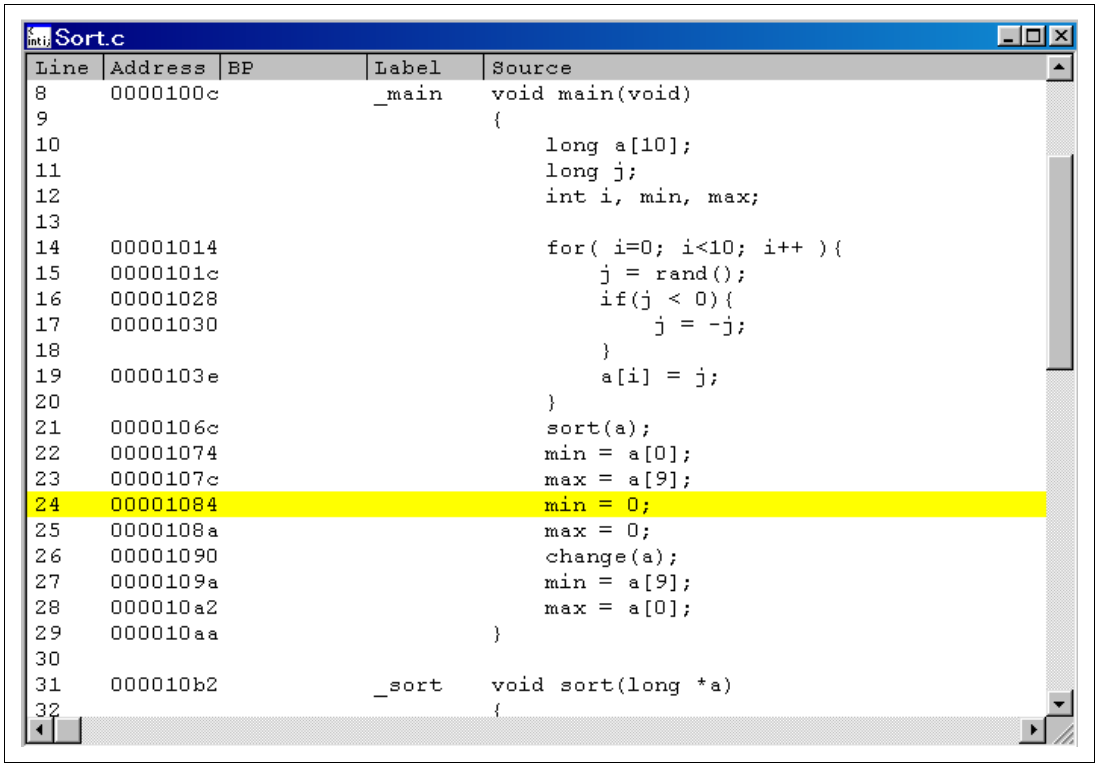


Figure 3.45 [Program] Window at Execution Stop (Break condition 1)

The [System Status] window displays the following contents.

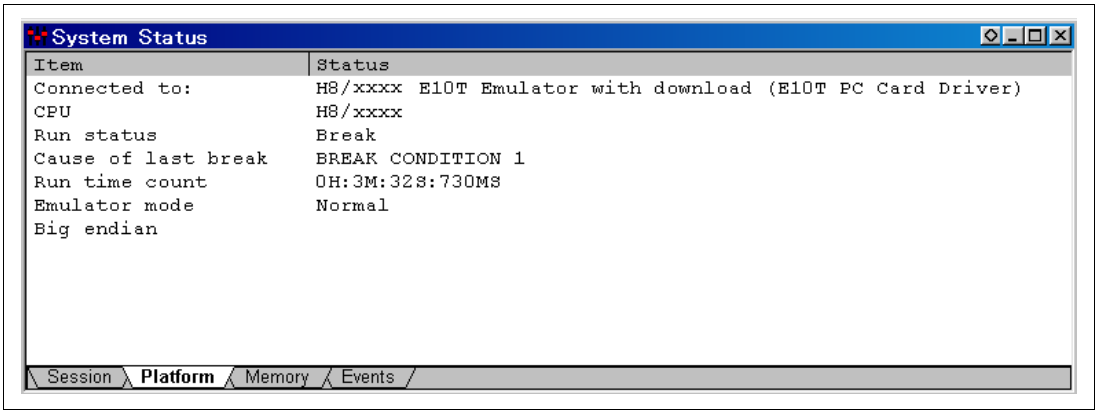


Figure 3.46 Displayed Contents of the [System Status] Window (Break condition 1)

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the on-line help.

3.17 Trace Function

The trace function of the E10T emulator is described.

The branch source addresses, mnemonics, operands, and source lines are displayed. Since this function uses the trace buffer built into the MCU, a realtime trace can be acquired.

Run the program as shown in the example of section 3.15.1, Software Break Function. The trace results are displayed in the [Trace] window after the program execution is completed.

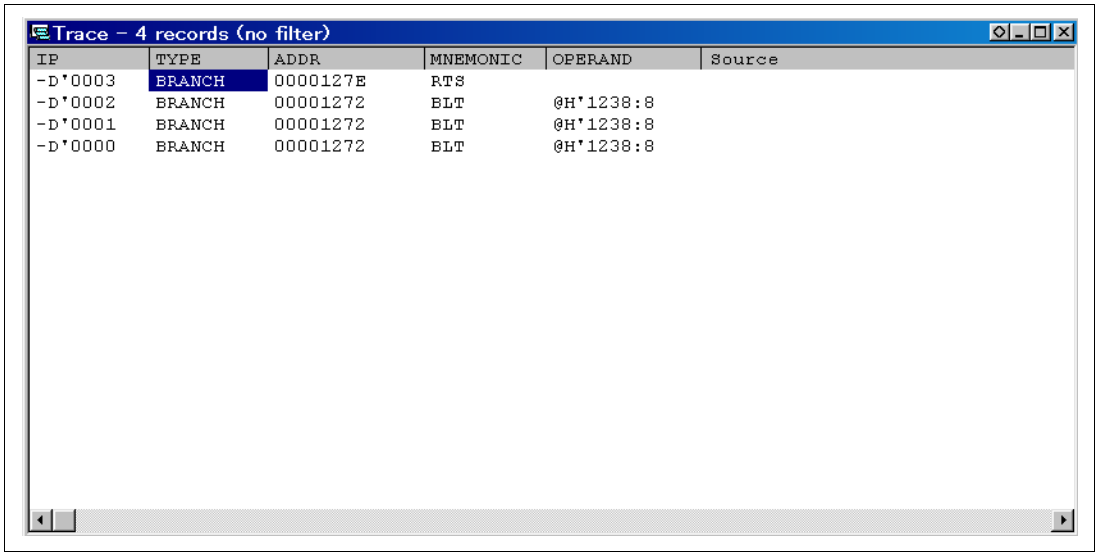

Trace – 4 records (no filter)
IP TYPE ADDR MNEMONIC OPERAND Source
-D'0003 BRANCH 0000127E RTS
-D'0002 BRANCH 00001272 BLT @H'1238:8
-D'0001 BRANCH 00001272 BLT @H'1238:8
-D'0000 BRANCH 00001272 BLT @H'1238:8

Figure 3.47 [Trace] Window

- If necessary, adjust the column width by dragging the header bar immediately below the title bar.

Note: The number of branch instructions that can be acquired by a trace differs according to the product. For the number that can be specified for each product, refer to the on-line help.

3.18 What Next?

This tutorial has described the major features of the emulator and the use of the HDI.

Sophisticated debugging can be carried out by using the emulation functions that the emulator offers. This provides for effective investigation of hardware and software problems by accurately isolating and identifying the conditions under which such problems arise.

Further details on the use of the HDI can be found in the separately issued Hitachi Debugging Interface User's Manual.

Section 4 Descriptions of Windows

4.1 HDI Windows

HDI window menu bars and the corresponding pull-down menus are listed in table 4.1. Where a description of a menu is included in the Hitachi Debugging Interface User's Manual or in this manual, an O mark or the relevant section number is shown. Related commands in the E10T Emulator User's Manual are also shown.

Table 4.1 HDI Window Menus and Related Manual Entries

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
File menu	New Session…	O	—
	Load Session…	O	—
	Save Session	O	2.6
	Save Session As…	O	—
	Load Program…	O	3.6.1
	Initialize	O	—
	Exit	O	—
Edit Menu	Cut	O	—
	Copy	O	—
	Paste	O	—
	Find…	O	—
	Evaluate…	O	—

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
View Menu	Breakpoints	O	3.10, 3.15.1, 4.2.3, 6.4.3
	Command Line	O	—
	Disassembly...	O	—
	I/O Registers	O	—
	Labels	O	—
	Locals	O	3.14
	Memory...	O	3.11
	Performance Analysis	O	—
	Profile-List	O	—
	Profile-tree	O	—
	Registers	O	3.8
	Source…	O	3.6.2
	Status	O	3.9, 3.15.1, 4.2.10
	Trace	O	4.2.8, 6.4.5, 6.4.6
	Watch	O	3.12
	Localized Dump Window	O	—
Run Menu	Reset CPU	O	—
	Go	O	3.9
	Reset Go	O	—
	Go to Cursor	O	—
	Set PC To Cursor	O	—
	Run…	O	—
	Step In	O	3.13.1
	Step Over	O	3.13.3
	Step Out	O	3.13.2
	Step…	O	—
	Halt	O	—

Table 4.1 HDI Window Menus and Related Manual Entries (cont)

Menu Bar	Pull-Down Menu	Hitachi Debugging Interface User's Manual	This Manual
Memory Menu	Refresh	O	—
	Load	O	—
	Save	O	—
	Verify	O	—
	Test	O	—
	Fill	O	—
	Copy	O	—
	Compare	O	—
Setup Menu	Status bar	O	—
	Options	O	—
	Radix	O	—
	Customise	O	—
	Configure Platform…	O	3.5, 4.2
Window Menu	Cascade	O	—
	Tile	O	—
	Arrange Icons	O	—
	Close All	O	—
Help Menu	Index	O	—
	Using Help	O	—
	Search for Help on	O	—
	About HDI	O	—

4.2 Descriptions of Each Window

This section describes each window. Figures in this section are used as examples. Each E10T emulator type has explanatory notes. Read section 6, H8/xxxx E10T Emulator Specifications.

4.2.1 [Configuration] Dialog Box

Function:

This dialog box sets the emulation conditions of the emulator.

Window:

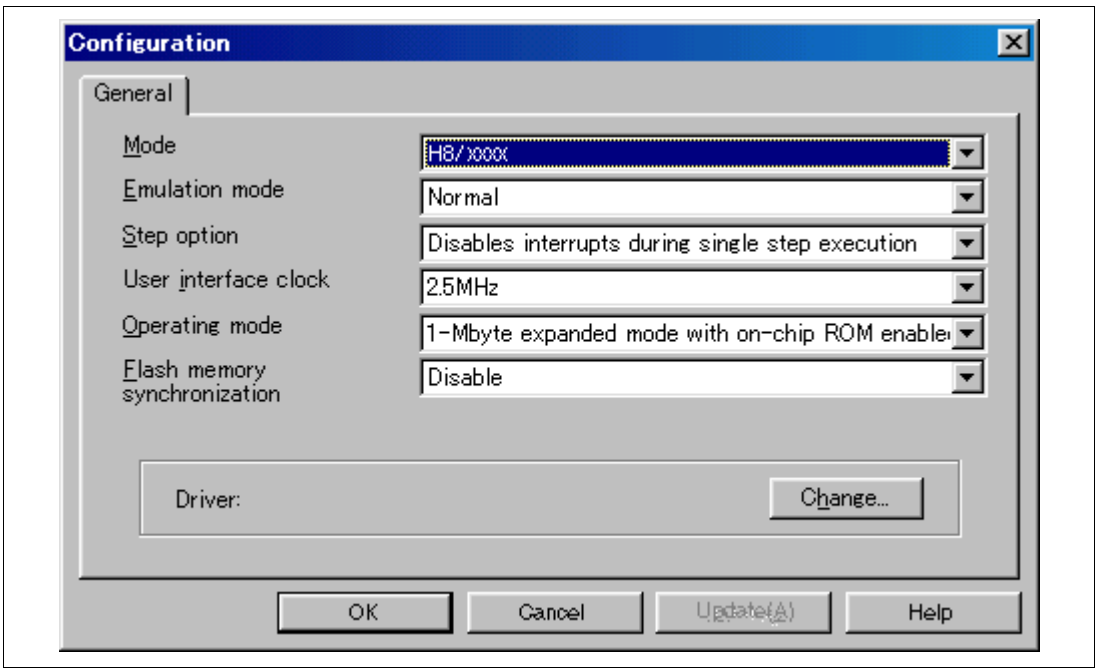


Figure 4.1 [Configuration] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Description:

The [Configuration] dialog box consists of the [General] page listed in table 4.2.

Table 4.2 [Configuration] Dialog Box Page

Page Name	Description
[General]	Sets the emulator operation conditions.

Clicking the [OK] button sets the emulation conditions. If the [Cancel] button is clicked, this dialog box is closed without setting the emulation conditions.

(1) [General] Page ([Configuration] Dialog Box)

Function:

This page sets the operational conditions for the emulator. The MCU name is displayed, the user interface clock is displayed, the emulation mode and interrupts during step execution are set, and the driver is selected.

Window:

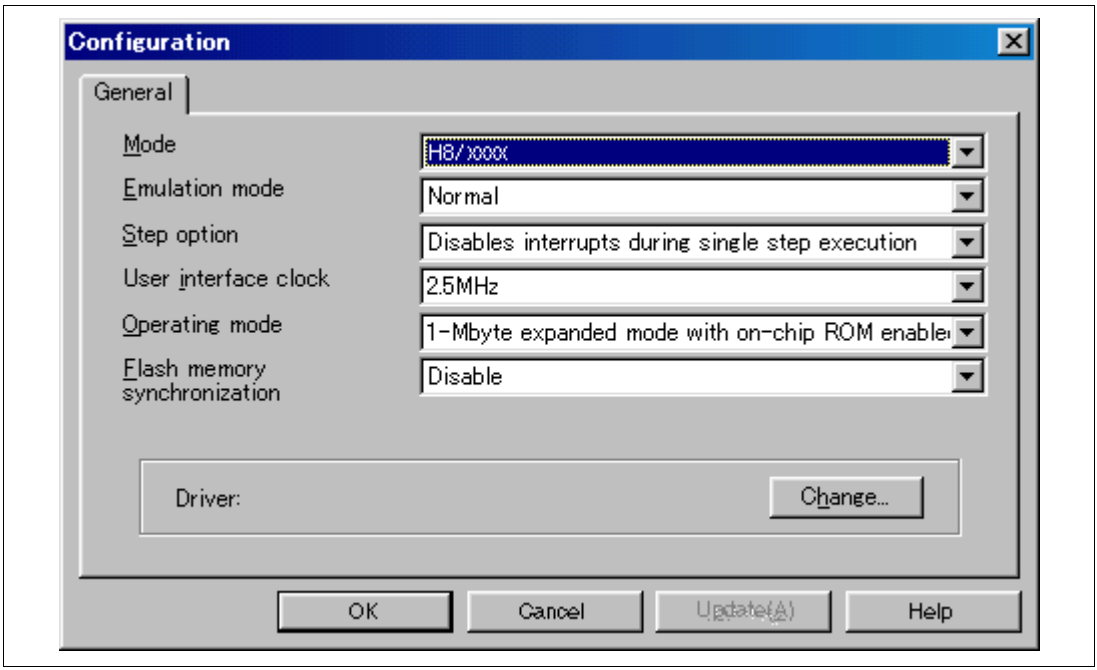


Figure 4.2 [General] Page ([Configuration] Dialog Box)

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Description:

Table 4.3 [General] Page Options

Option	Description
[Mode] combo box	Displays the MCU name.
[Emulation mode] combo box	Selects the emulation mode at user program execution. Select Normal to perform normal emulation. Select No Break to disable breakpoint settings.
[Step option] combo box	Enables or disables interrupts during step execution. Disables interrupts during single step execution: Interrupts during step execution are masked. Enables interrupts during single step execution: Interrupts during step execution are released.
[User interface clock] combo box	Displays the transfer clock of the user interface.
[Operating mode] combo box	Displays the CPU operating mode set by the MD pin. 1-Mbyte expanded mode with on-chip ROM enabled: For mode 5 16-Mbyte expanded mode with on-chip ROM enabled: For mode 6

Table 4.3 [General] Page Options (cont)

Option	Description
[Flash memory synchronization] combo box	Synchronization method is set between the host computer and the flash memory. When synchronization is performed from the host computer to the flash memory, a waiting time will be generated to write the flash memory during user program halting, but the displayed contents and the flash memory are always matched. When synchronization is performed from the flash memory to the host computer, the rewritten contents in the user program mode will be reflected since the flash memory is read during user program halting. Disable: Synchronization is not performed except when the E10A emulator is activated and the flash memory area is modified. PC to Flash memory: Synchronization is performed from the host computer t the flash memory. Flash memory to PC: Synchronization is performed from the flash memory to the host computer. PC to Flash memory, Flash memory to PC: Synchronization is performed between the host computer and the flash memory.
[Driver] group box	The [E10T Driver Details] dialog box is displayed. Changes the driver currently connected.

When a driver is to be changed with the [Change…] button, the following message is displayed.

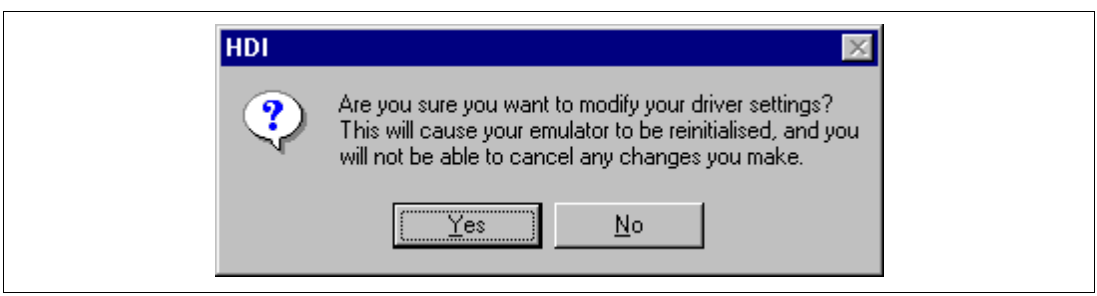

HDI
Are you sure you want to modify your driver settings?
This will cause your emulator to be reinitialised, and you
will not be able to cancel any changes you make.
Yes
No

Figure 4.3 Warning Message Box

When the [Yes] button is clicked, the [E10T Driver Details] dialog box is displayed. When the [No] button is clicked, the display returns to the [Configuration] dialog box.

Related Items:

[Configuration] dialog box
GO_OPTION command
STEP_INTERRUPT command

4.2.2 [E10T Driver Details] Dialog Box

When the [OK] button is clicked, the [E10T Driver Details] dialog box will appear to select the driver software used by the interface to the emulator. Since the emulator is reinitialized after this dialog box is closed, it cannot be cancelled.

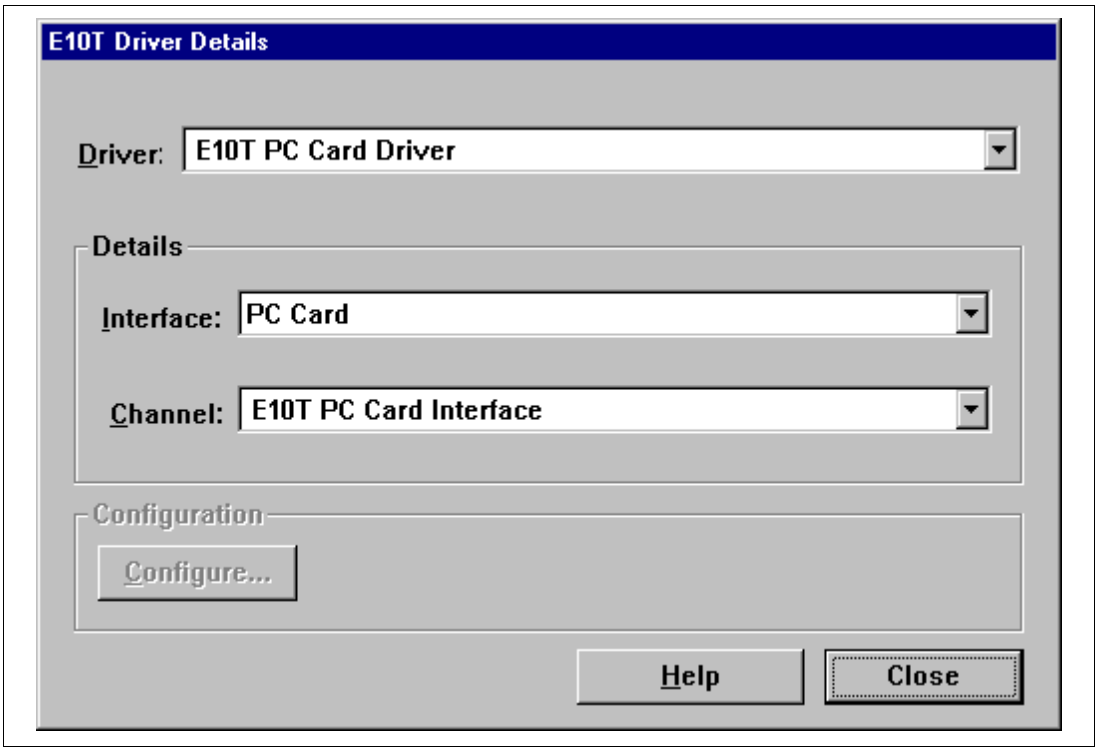


Figure 4.4 [E10T Driver Details] Dialog Box

Table 4.4 shows each option.

Table 4.4 [E10T Driver Details] Dialog Box Option

Option	Description
[Driver] combo box	Selects the driver to connect the HDI with the emulator.
[Interface] combo box	Selects the interface name of the card emulator to be connected.
[Channel] combo box	Selects the interface that the PC interface board has been connected.
[Configure...] button	Clicked to display that the driver supports the configuration dialog box.

When the PCMCIA card emulator is used, [E10T PC Card Driver] is selected. When the PCI card emulator is used, [E10T PCI Card Driver] is selected.

Note: When the HDI is not linked up even if the above procedure has been executed, the driver may not be set correctly. Install drivers provided with the /SETUP directory in the CD-R according to the screen instructions.

Related Items:

[Configuration] dialog box
[General] page

4.2.3 [Breakpoints] Window

Function:

This window lists all break conditions that have been set.

Window:

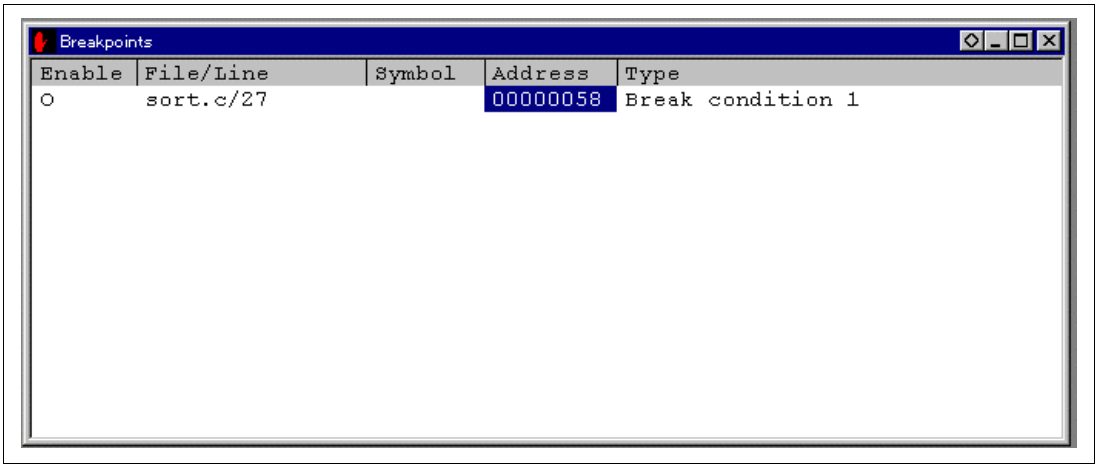


Figure 4.5 [Breakpoints] Window

Description:

The [Breakpoints] window displays breakpoint setting information. The items listed in the following tables are displayed.

Table 4.5 [Breakpoints] Window Display Items

Item	Description
[Enable]	Indicates whether the break condition is enabled or disabled. Symbol ● or ❍ shows that the break condition is enabled.
[File/Line]	Displays the file name and line number where the breakpoint is set.
[Symbol]	Displays the symbol corresponding to the breakpoint address. If no symbol has been defined for the address, a blank is displayed.
[Address]	Displays the address where the breakpoint is set.
[Type]	Displays the break condition type as follows: Breakpoint: Software breakpoint Break condition n: Hardware break condition (n is a number)

The pop-up menu, which is opened by clicking the right mouse button, can be used to set, change, and clear breakpoints, and to enable or disable break conditions. The pop-up menu functions are described in the following table.

Table 4.6 [Breakpoints] Window Pop-up Menu Operation

Menu Name	Description
[Add]	Sets break conditions. Clicking this button will display the [Break] dialog box, enabling break conditions to be set.
[Edit]	Changes break conditions. Select break conditions to be changed and click this button. The break condition setting dialog box will be displayed, enabling the break condition to be changed.
[Disable] ([Enable])	Enables or disables break conditions. Select break conditions to be enabled or disabled and click this button.
[Delete]	Clears break conditions. Select break conditions to be cleared and click this button.
[Delete All]	Clears all break conditions.
[Go to Source]	Jumps to the address which sets the break in the [Source] window.

4.2.4 [Break] Dialog Box

Function:

This dialog box displays the break condition settings.

Window:

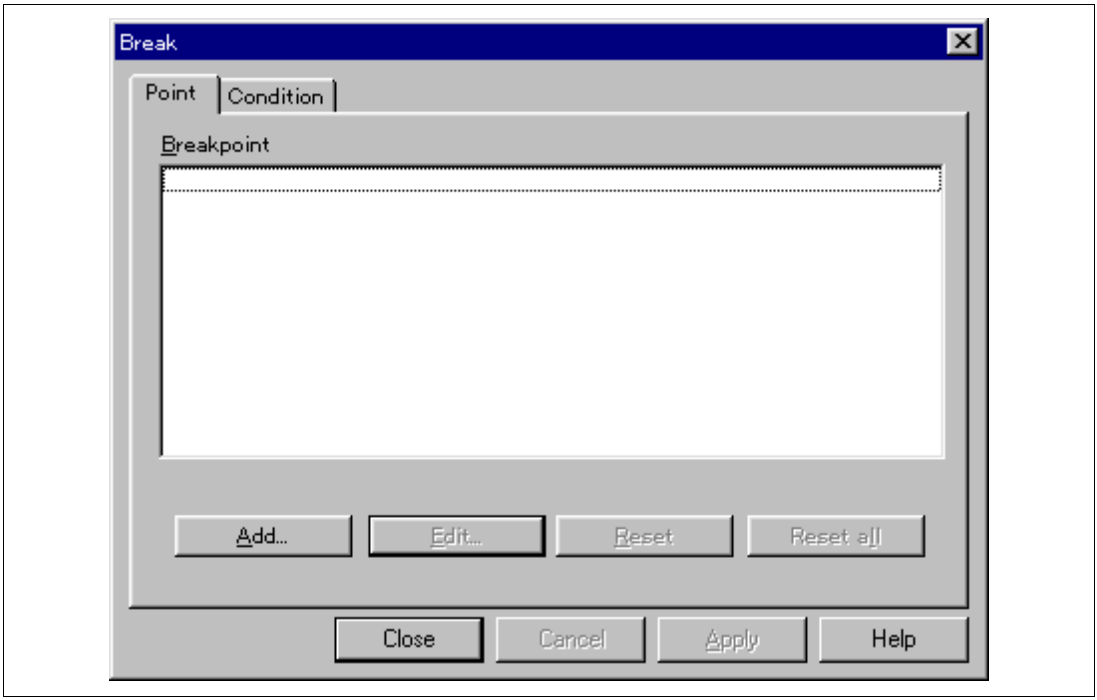


Figure 4.6 [Break] Dialog Box

Description:

The [Break] dialog box consists of the pages listed in table 4.7.

Table 4.7 [Break] Dialog Box Pages

Page Name	Description
[Point]	Displays software breakpoint settings.
[Condition]	Displays Break Condition settings.

The dialog boxes which set or modify break conditions can be displayed from the pages above.

Clicking the [Close] button will close this dialog box. The [Apply] button is not supported.

(1) [Point] Page ([Break] Dialog Box)

Function:

This page displays software breakpoint settings. In this page, software breakpoints can be set, changed, and cleared.

Window:

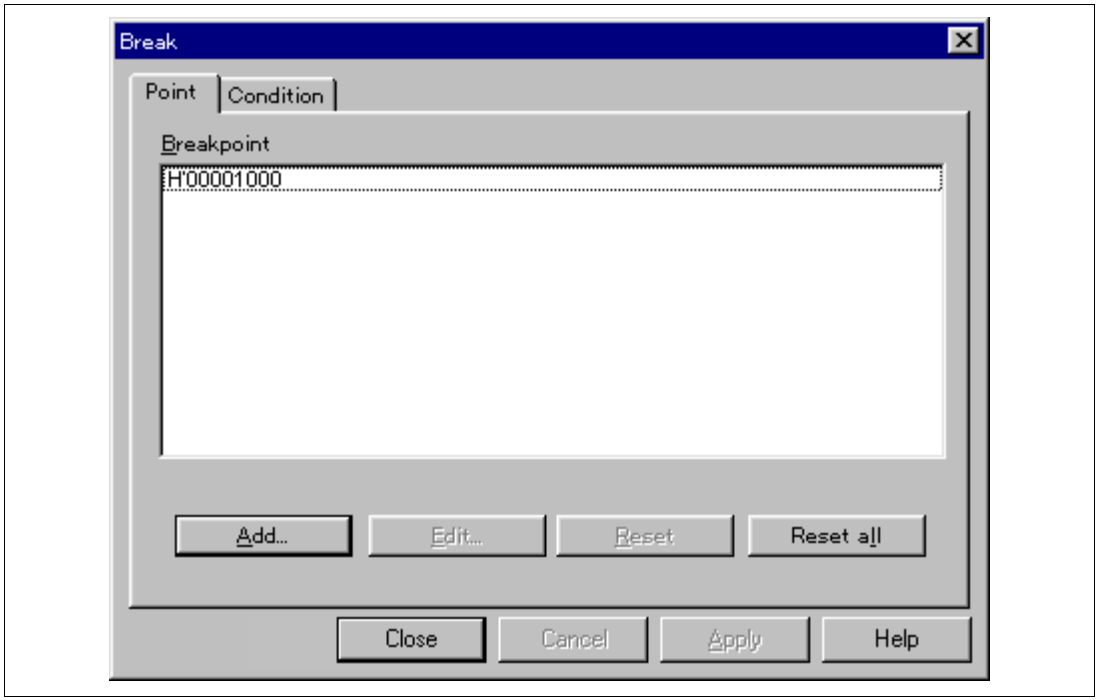


Figure 4.7 [Point] Page ([Break] Dialog Box)

Description:

Table 4.8 [Point] Page Options

Option	Description
[Breakpoint] list box	Lists the software breakpoints currently being set. The display contents are as follows: <breakpoint address>
[Add...] button	Sets software breakpoints. Clicking this button displays the [Breakpoint] dialog box.
[Edit...] button	Changes the software breakpoint selected in the [Breakpoint] list box. Clicking this button displays the [Breakpoint] dialog box.
[Reset] button	Clears the software breakpoint selected in the [Breakpoint] list box.
[Reset all] button	Clears all software breakpoints displayed in the [Breakpoint] list box.

Related Items:

[Breakpoint] dialog box
BREAKPOINT command
BREAKPOINT_CLEAR command
BREAKPOINT_DISPLAY command
BREAKPOINT_ENABLE command

(2) [Condition] Page ([Break] Dialog Box)

Function:

This page displays the Break Condition settings. These conditions can also be set or cleared in this page.

Window:

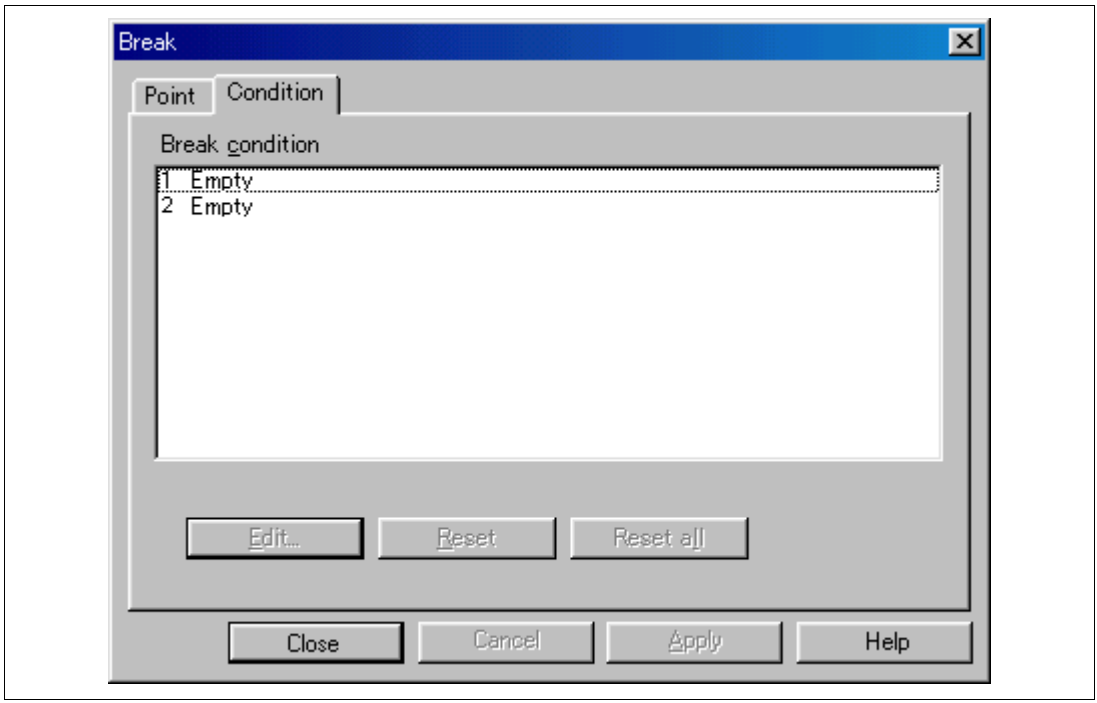


Figure 4.8 [Condition] Page ([Break] Dialog Box)

Note: Note that the number of hardware break conditions differs according to the product. For the number that can be specified for each product, refer to the on-line help.

Description:

Table 4.9 [Condition] Page Options

Option	Description
[Break condition] list box	Displays the Break condition settings. The display at system initiation is as follows: When conditions are set, Enable is displayed. When no conditions are set, Empty is displayed. 1 Empty (setting of Break Condition 1) 2 Empty (setting of Break Condition 2) :
[Edit...] button	Changes the Break condition settings selected in the [Break condition] list box. Clicking this button displays the [Break condition] dialog boxes.
[Reset] button	Clears the Break condition settings selected in the [Break condition] list box.
[Reset all] button	Clears all Break condition settings in the [Break condition] list box.

Related Items:

[condition] page
BREAKCONDITION_CLEAR command
BREAKCONDITION_DISPLAY command
BREAKCONDITION_ENABLE command
BREAKCONDITION_SET command

4.2.5 [Breakpoint] Dialog Box

Function:

This dialog box sets software breakpoints.

Window:

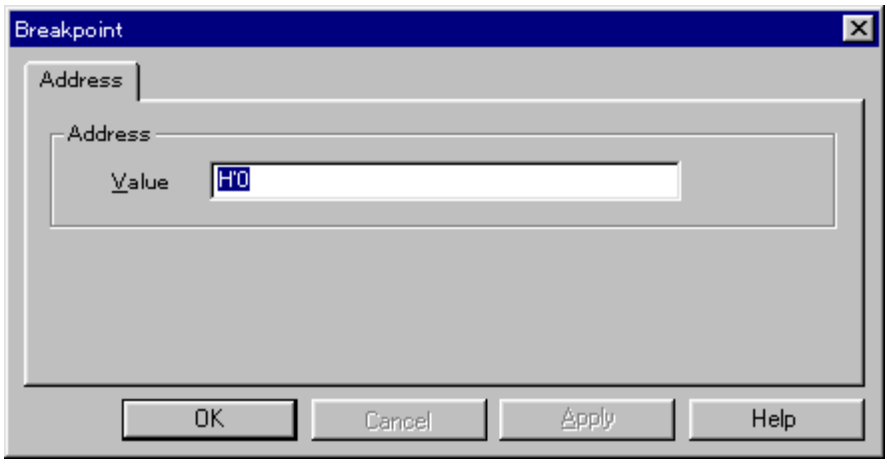


Figure 4.9 [Breakpoint] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Description:

The [Address] page options are as follows:

Table 4.10 [Address] Page Options

Option	Description
[Value] edit box	Sets a breakpoint address with a number or a symbol.

Clicking the [OK] button enables breakpoints to be set. If the [Cancel] button is clicked, this dialog box is closed without setting breakpoints.

Related Items:

[Point] page
BREAKPOINT command
BREAKPOINT_CLEAR command
BREAKPOINT_DISPLAY command
BREAKPOINT_ENABLE command

4.2.6 [Break condition] Dialog Box

Function:

This dialog box sets hardware break conditions.

Window:

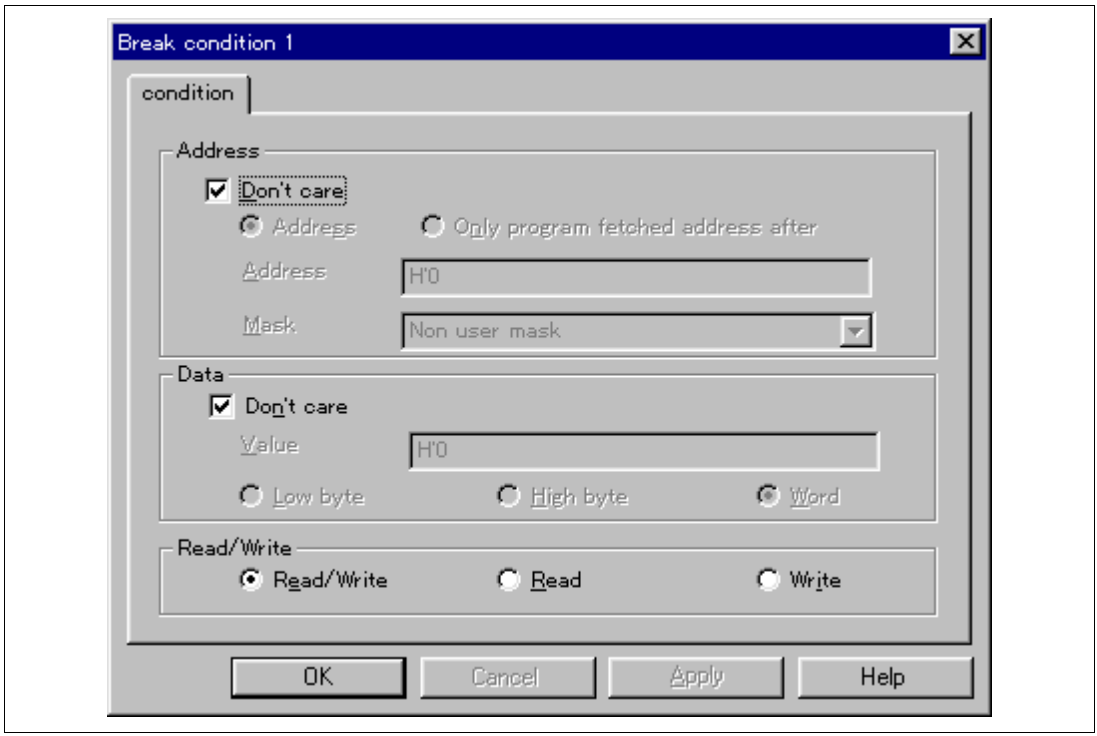


Figure 4.10 [Break condition 1] Dialog Box

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Description:

The [Break condition] dialog box consists of the [condition] page. Sets the conditions for halting the program.

Contents to be set by this page are described in section 4.2.7, [Break condition] Dialog Box Pages.

Clicking the [OK] button sets the hardware break conditions. If the [Cancel] button is clicked, the dialog box is closed without setting the hardware break conditions.

Related Items:

[condition] page
BREAKPOINT command
BREAKCONDITION_CLEAR command
BREAKCONDITION_DISPLAY command
BREAKCONDITION_ENABLE command

4.2.7 [Break condition] Dialog Box Pages

Function:

The [Break condition] dialog box page allows a number of hardware break conditions to be set. Some functions may not be supported by some types of emulators. The setting conditions may differ from the dialog box name in table 4.11. For details, refer to section 6.4.2, Hardware Break Functions.

Table 4.11 Setting Conditions in [Break condition] Dialog Box

	Type		
Dialog Box	**Address Bus Condition**	**Data Bus Condition**	**Read/Write Condition**
[Break condition 1] dialog box	O	O	O
[Break condition 2] dialog box	O	x	x

Note: O: Can be set by checking the radio button in the dialog box.

Table 4.12 shows the [Break condition] dialog box page.

Table 4.12 [Break condition] Dialog Box Pages

Page Name	Function
[condition]	Sets the address bus, data bus, and read/write cycle conditions of Break condition 1, and the address bus condition of Break condition 2.

Note: This function differs according to the product. For the specifications of each product, refer to section 6.4.2, Hardware Break Functions, or to the on-line help.

(1) [condition] Page

Function:

This page sets the address bus, data bus, and read/write cycle conditions.

Window:

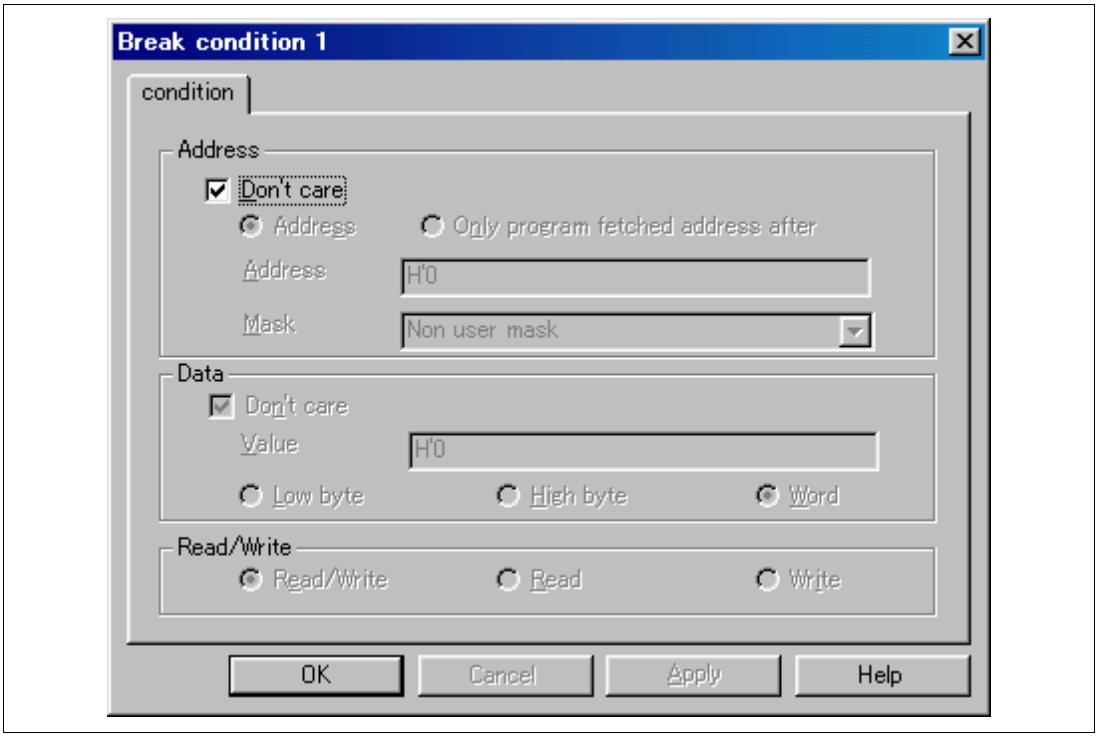


Figure 4.11 [condition] Page

Note: The items that can be set in this window differ according to the product. For the settings for each product, refer to the on-line help.

Description:

The [Address] group box sets the address bus conditions.

Table 4.13 [Address] Group Box Options

Option	Description
[Don't care] check box	Does not set address conditions.
[Address] radio button	Sets use of the normal address bus as break conditions.
[Only program fetched address after] radio button	Sets a break after prefetched address execution as break conditions.
[Address] edit box	Sets the address bus value with a number or a symbol.
[Mask] list box	Sets the mask bits. For masked bits, the break condition is satisfied regardless of the address values. To disable the setting, select Non user mask. Lower 4bit: The lower 4 bits are masked. Lower 8bit: The lower 8 bits are masked. Lower 12bit: The lower 12 bits are masked. Lower 16bit: The lower 16 bits are masked. Lower 20bit: The lower 20 bits are masked.

The contents of an option that can be set will change depending on the radio button selected.

Table 4.14 Radio Button Options

Option	Description
[Address] radio button	All conditions can be set.
[Only program fetched address after] radio button	The [Address] condition can be set.

The [Data] group box sets the data bus conditions.

Table 4.15 [Data] Group Box Options

Option	Description
[Don't care] check box	No data conditions are set.
[Value] edit box	Sets the data bus with a number.
[Low byte] radio button	Sets data access size as the lower 8-bit access.
[High byte] radio button	Sets data access size as the upper 8-bit access.
[Word] radio button	Sets data access size as the word access.

The [Read/Write] group box sets the read/write cycle conditions.

Table 4.16 [Read/Write] Group Box Options

Option	Description
[Read/Write] radio button	Sets the read/write cycle conditions as break conditions.
[Read] radio button	Sets read cycles as break conditions.
[Write] radio button	Sets write cycles as break conditions.

Related Items:

[Condition] page
BREAKCONDITION_CLEAR command
BREAKCONDITION_DISPLAY command
BREAKCONDITION_ENABLE command
BREAKCONDITION_SET command

4.2.8 [Trace] Window

Function:

This window displays the trace buffer contents. The contents are displayed in both C and assembly languages to simplify debugging.

Window:

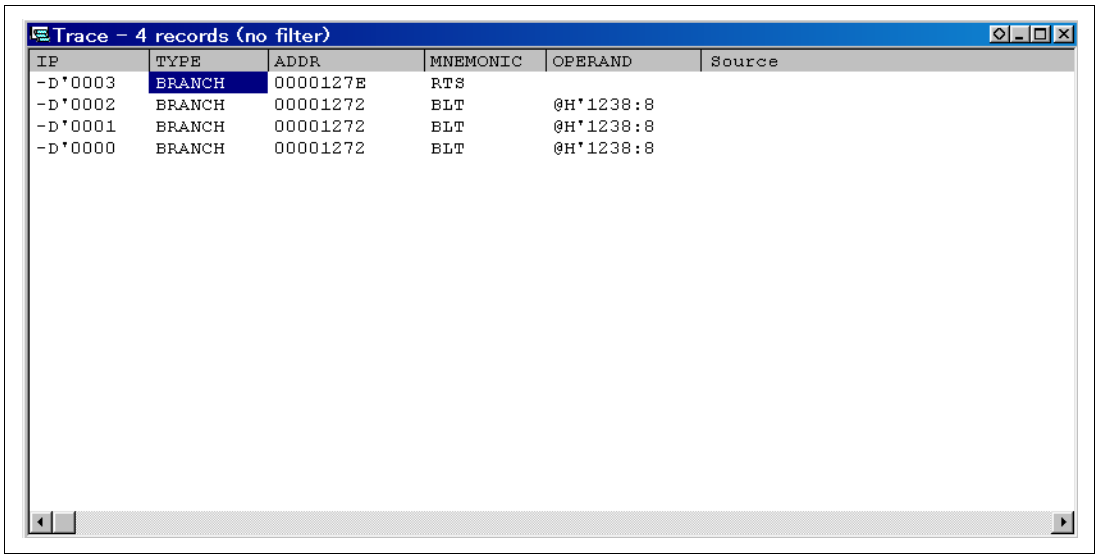

Trace – 4 records (no filter)
IP TYPE ADDR MNEMONIC OPERAND Source
-D'0003 BRANCH 0000127E RTS
-D'0002 BRANCH 00001272 BLT @H'1238:8
-D'0001 BRANCH 00001272 BLT @H'1238:8
-D'0000 BRANCH 00001272 BLT @H'1238:8

Figure 4.12 [Trace] Window

Note: The number of branch instructions that can be acquired by a trace differs according to the product. For the number that can be specified for each product, refer to the on-line help.

Description:

This window displays the trace buffer contents. The items listed in table 4.17 are displayed.

Table 4.17 [Trace] Window Display Items

Item	Description
[IP]	Displays the instruction pointer (signed decimal).
[TYPE]	Displays the branch source. BRANCH: Branch source
[ADDR]	Displays the branch source address.
[MNEMONIC]	Displays the execution instruction mnemonic.
[OPERAND]	Displays the execution instruction operand.
[Source]	Displays the C-source line of the address that the trace has been acquired.
[Total Records]	Displays the total number of instructions displayed as trace information in the [Trace] window.

The pop-up menu, opened by clicking the right mouse button, can be used to set, change, and clear trace conditions. For details, refer to the Hitachi Debugging Interface User's Manual.

Note: The [Halt] menu in the pop-up menu is active only when the [Trace] window is open during user program execution. Realtime emulation cannot be performed by using the [Halt] menu.

Related Item:

TRACE_DISPLAY command

4.2.9 [System Status] Window

Function:

This window lists information, such as conditions that have been set to the emulator and execution results.

Window:

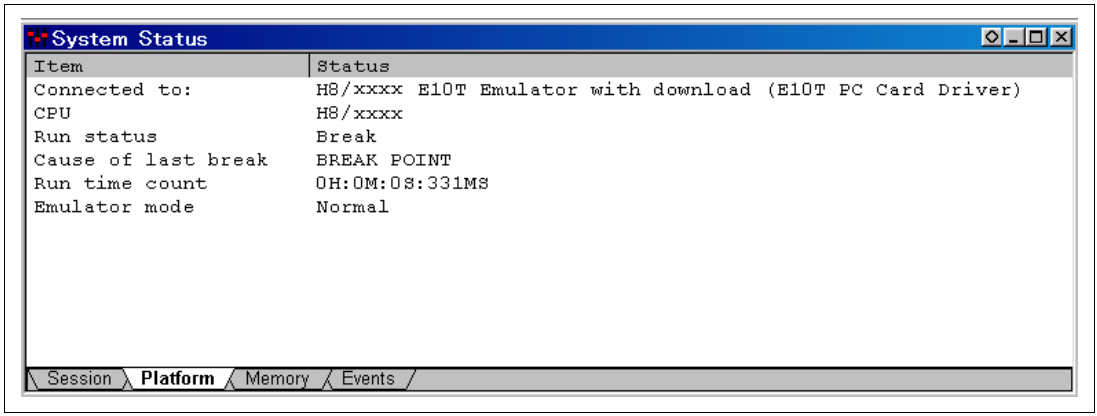


Figure 4.13 [System Status] Window

Note: The items that can be displayed in this window differ according to the product. For the items that can be displayed, refer to the on-line help.

Description:

The items listed in the following table are displayed in the [System Status] window.

Table 4.18 [System Status] Window Display Items

Page	Item	Description
[Session]	Target System	Displays whether the emulator is connected.
	Session Name	Displays the session file name.
	Program Name	Displays the load module file name.
[Platform]	Connected To	Displays the name of the connected emulator and the selected driver name.
	CPU	Displays the target MCU name.
	Run status	Displays the execution status: Running: Being executed Break: Stopped
	Cause of last break	Displays the cause of the emulator stopping at break.
	Run time count	Displays the program execution time. The display format is H: hours, M: minutes, S: seconds, and MS: milliseconds. In this example, 0H:0M:0S:331MS is displayed.
	Emulator mode	Displays the emulator operating mode (setting information for [Emulation Mode] of the [Configuration] dialog box).
[Memory]	Loaded Memory Areas	Displays the loaded area of the load module.
[Events]	Resources	Displays the usage states of software breakpoint and Break condition.

Related Items:

[Configuration] dialog box
STATUS command

Section 5 Command-line Functions

5.1 Table and Symbol Description

This section describes the format used in section 5.2, Command Descriptions. The descriptions of some commands are given over two or more pages.

5.1.1 Format

The input format for each command is as follows. Characters shown in bold-italics are to be input.

[ ] : Parameters enclosed by [ ] can be omitted.

< > : Contents shown in < > are set.

< >=: The parameter to the left of the "=" sign is input in the format shown to the right.

| : This represents a non-exclusive selection.

| | : This represents an exclusive selection.

The command parameter details are described in the parameter table.

5.1.2 Parameter Input

Numerical Parameters:

A binary, octal, decimal, or hexadecimal value, a symbol, or a formula can be input. A symbol can contain up to 32 characters. Terms in a formula are separated with operators (such as + or –).

Keyword Parameters:

One of the bold characters given in the description column of the table can be input. If a character string not shown in the description is input, an error will occur.

Character-String Parameters:

Character-string parameters are used to input mask data or a file name. In the mask data, set a radix (H': hexadecimal or B': binary) at the top of a character string and set * at the digit to be masked.

5.1.3 Examples

These are actual input examples. For commands whose execution results in a specific display output, an example of the display is given.

5.1.4 Related Items

Related E10T HDI commands (abbreviations) and dialog boxes are shown. (Refer to section 4, Descriptions of Windows.)

5.2 Command Descriptions

The command list of the E10T emulator is shown below.

Table 5.1 E10T HDI Commands

No	Command	Abb.	Function
1	BREAKCONDITION_ CLEAR	BCC	Clears hardware breakpoints that have been set.
2	BREAKCONDITION_ DISPLAY	BCD	Displays hardware breakpoints that have been set.
3	BREAKCONDITION_ ENABLE	BCE	Enables or disables hardware breakpoints that have been set.
4	BREAKCONDITION_ SET	BCS	Sets hardware breakpoints.
5	BREAKPOINT	BP	Sets software breakpoints.
6	BREAKPOINT_CLEAR	BC	Clears software breakpoints that have been set.
7	BREAKPOINT_DISPLAY	BD	Displays software breakpoints that have been set.
8	BREAKPOINT_ENABLE	BE	Enables or disables software breakpoints that have been set.
9	DEVICE_TYPE	DE	Displays MCU type currently selected.
10	GO_OPTION	GP	Sets or displays the emulation mode during user program execution.
11	REFRESH	RF	Updates the HDI memory information to the latest contents.
12	RESET	RE	Resets the H8/xxxx.
13	STATUS	STS	Displays emulator status information.
14	STEP_INTERRUPT	SI	Sets or displays the enable or disable status of interrupts during step execution.
15	TRACE_DISPLAY	TD	Displays acquired trace information.

Note: Support for these commands varies with the product. For the specifications of each product, refer to the on-line help.

5.2.1 BREAKCONDITION_CLEAR: BCC

Description:

Clears hardware breakpoints that have been set.

Format:

```
bcc [<channel>]
```

<channel> = channel <channel_number>

Table 5.2 BREAKCONDITION_CLEAR Command Parameter

Parameter	Type	Description
<channel number>	Numerical value	Specifies hardware break channel number

Note: When <channel> is omitted, all hardware breakpoints that have been set are cleared.

Examples:

To clear all hardware breakpoints:

```
>bcc (RET)
```

To clear a hardware breakpoint set at channel 2:

```
>bcc channel 2 (RET)
```

Related Items:

BCD, BCE, and BCS commands
[Breakpoints] window
[Break] and [Break Condition] dialog boxes

5.2.2 BREAKCONDITION_DISPLAY: BCD

Description:

Displays hardware breakpoints that have been set. The display contents include a hardware breakpoint channel number, enable or disable of the setting, and setting conditions.

Format:

```
bcd [<channel>]
```

<channel> = channel <channel_number>

Table 5.3 BREAKCONDITION_DISPLAY Command Parameter

Parameter	Type	Description
<channel_number>	Numerical value	Hardware breakpoint channel number

Note: When <channel> is omitted, all hardware breakpoints that have been set are displayed.

Examples:

To display all hardware breakpoint settings:

```
>bcd (RET)
```

Break Condition 1:Enable data 20 long
Break Condition 2:Disable address 126

To display the hardware breakpoint set at channel 1:

```
>bcd channel 1 (RET)
Break condition 1: Enable address H'1000 data H'55 lbyte direction read
```

Note: The items displayed with this command differ according to the product. For the display specifications of each product, refer to the online help.

Related Items:

BCC, BCE, and BCS commands
[Breakpoints] window
[Break] and [Break Condition] dialog boxes

5.2.3 BREAKCONDITION_ENABLE: BCE

Description:

Enables or disables hardware breakpoints that have been set.

Format:

```
bce [<channel>] <mode>
```

<channel> = channel <channel_number>

Table 5.4 BREAKCONDITION_ENABLE Command Parameters

Parameter	Type	Description
<channel_number>	Numerical value	Specifies hardware break channel number.
<mode>	Keyword	Enables or disables hardware break settings. Set either of the following keywords: **enable**: Enables hardware break settings. **disable**: Disables hardware break settings.

Note: When <channel> is omitted, all hardware breakpoints that have been set are enabled or disabled.

Examples:

To enable all hardware breakpoints:

```
>bce enable (RET)
```

To disable the hardware breakpoints set at channel 1:

```
>bce channel 1 disable (RET)
```

Related Items:

BCC, BCD, and BCS commands
[Breakpoints] window
[Break] and [Break Condition] dialog boxes

5.2.4 BREAKCONDITION_SET: BCS

Description:

Sets hardware breakpoints.

Note: The function will be different according to the MCUs used. For functions of each emulator product, refer to section 6.4.2, Hardware Break Functions.

Format:

```
bcs <channel>  <option> [<option> ... ]
```

<option> = <addropt> | <dataopt> | <r/wopt> |
<addropt> = address <address> [<addrcycle>] [<addr_mask>]
<dataopt> = data <data> <size>
<r/wopt> = direction <r/w>

Table 5.5 BREAKCONDITION_SET Command Parameters

Parameter	Type	Description
<address>	Numerical value, character string	Numerical value or symbol representing an address bus value
<addrcycle>	Keyword	Address bus access conditions **pcafter**: The address bus is targeted during program fetch cycles. Breaks after the instruction at the address set by the <address> parameter is executed. When <addrcycle> is omitted, the address in the program fetch cycles, and the address bus during data access cycles are targeted.
<addr_mask>	Keyword	Sets mask conditions for the address bus. The masked bits satisfy the condition regardless of the address values. Set either of the following: **m1**: The lower 4 bits are masked. **m2**: The lower 8 bits are masked. **m3**: The lower 12 bits are masked. **m4**: The lower 16 bits are masked. **m5**: The lower 20 bits are masked.
<data>	Numerical value	Data bus value
<size>	Keyword	Sets the access conditions of the data bus. Set either of the following: **lbyte**: Lower 8-bit byte access **hbyte**: Upper 8-bit byte access **word**: Word access
<r/w>	Keyword	Read/write conditions Set either of the following keywords: **read**: read cycles **write**: write cycles
<channel_number>	Numerical value	Hardware break condition channel number Specifiable options change depending on the channel number. For details, refer to section 6.4.2, Hardware Break Functions. **1**: <addropt>, <dataopt>, and <r/wopt> can be set. **2**: address <address> pcafter

Note: The parameters for this command vary with the product. For the specifications of each product, refer to the on-line help.

Examples:

To set the following conditions as a hardware breakpoint:

Address condition: An address bus value of H'1000,
Data condition: A data bus value of H'55 and lower 8-bit byte access,
Read/write cycle condition: Read cycle.

```
>bcs channel 1 address 1000 data 55 lbyte direction read (RET)
```

To set the following conditions as a hardware breakpoint:

Address condition: An address bus value of H'2000 and instruction fetch cycle.

```
>bcs channel 2 address 2000 pcafter (RET)
```

To set the following conditions as a hardware breakpoint:

Address condition: The lower 4 bits of H'1000 are masked,
Data condition: A data bus value of H'aa and upper 8-bit byte access.

```
>bcs channel 1 address 1000 m1 data aa hbyte (RET)
```

5.2.5 BREAKPOINT: BP

Description:

Sets software breakpoints.

Note: The function will be different according to the MCUs used.

Format:

```
bp <address>
```

Table 5.6 BREAKPOINT Command Parameters

Parameter	Type	Description
<address>	Numerical value, character string	Breakpoint value set by numerical value or symbol When an odd address is set, the address is rounded down to an even address.

Note: The parameters for this command vary with the product. For the specifications of each product, refer to the on-line help.

Examples:

To set a software breakpoint at address H'12c8:

```
>bp H'12c8 (RET)
```

5.2.6 BREAKPOINT_CLEAR: BC

Description:

Clears software breakpoints that have been set.

Format:

```
bc [<address>]
```

Table 5.7 BREAKPOINT_CLEAR Command Parameters

Parameter	Type	Description
<address>	Numerical value, character string	Breakpoint value set by numerical value or symbol

Note: When no parameters are set, all software breakpoints are cleared.

Note: The parameters for this command vary with the product. For the specifications of each product, refer to the on-line help.

Examples:

To clear all software breakpoints:

```
>bc (RET)
```

To clear all software breakpoints whose address value is H'1000:

```
>bc H'1000 (RET)
```

5.2.7 BREAKPOINT_DISPLAY: BD

Description:

Displays software breakpoints that have been set.

Format:

```
bd
```

Table 5.8 BREAKPOINT_DISPLAY Command Parameter

Parameter	Type	Description
None		

Example:

To display the software breakpoints that have been set:

```
>bd (RET)
```

The display format is as follows:

```
>bd
  H'00000110 Enable
  H'0000011c Disable
  H'00000250 Enable
```

Note: The items displayed with this command vary with the product. For the display specifications of each product, refer to the on-line help.

5.2.8 BREAKPOINT_ENABLE: BE

Description:

Enables or disables software breakpoints that have been set.

Format:

```
be <address> <mode>
```

Table 5.9 BREAKPOINT_ENABLE Command Parameters

Parameter	Type	Description
<address>	Numerical value, character string	Breakpoint value set by numerical value or symbol
<mode>	Keyword	Enables or disables breakpoints. Set either of the following: **enable**: Enables breakpoints. **disable**: Disables breakpoints.

Examples:

To enable software breakpoints that have been set:

```
>be H'1002 enable (RET)
```

To disable software breakpoints that have been set:

```
>be H'1002 disable (RET)
```

Notes: 1. The parameters for this command vary with the product. For the specifications of each product, refer to the on-line help.

2. The items displayed with this command vary with the product. For the display specifications of each product, refer to the on-line help.

5.2.9 DEVICE_TYPE: DE

Description:

Displays MCU type currently selected.

Format:

```
de
```

Table 5.10 DEVICE_TYPE Command Parameter

Parameter	Type	Description
None		

Example:

To display the currently selected MCU:

```
>de (RET)
```

The display format is as follows:

```
>de
Current device = H8/xxxx
```

5.2.10 GO_OPTION: GP

Description:

Sets or displays the emulation mode during user program execution.

Format:

```
gp
```

Displays emulation mode for user program execution.

```
gp <eml_opt>
<eml_opt> = eml_mode <eml_mode>
```

Sets emulation mode for user program execution.

Table 5.11 GO_OPTION Command Parameter

Parameter	Type	Description
<eml_mode>	Keyword	Specifies the emulation mode. Set either of the following: **normal**: Normal execution **no_break**: Makes software breakpoints and hardware breakpoints temporarily invalid and executes the user program.

Note: The parameters for this command vary with the product. For the specifications of each product, refer to the on-line help.

Examples:

To display the currently set emulation mode for user program execution:

```
>gp (RET)
```

The display format is as follows:

```
>gp
Emulator execution mode = Normal
```

To set the normal emulation mode for user program execution:

```
>gp eml_mode normal (RET)
```

Note: The items displayed with this command vary with the product. For the display specifications of each product, refer to the on-line help.

5.2.11 REFRESH: RF

Description:

Updates the HDI memory information to the latest contents.

Format:

```
rf
```

Table 5.12 REFRESH Command Parameter

Parameter	Type	Description
None		

Example:

To update the HDI memory information:

```
>rf (RET)
```

5.2.12 RESET: RE

Description:

Resets the H8/xxxx. The breakpoint settings are retained. After the reset, the registers will be in the same state as when power is supplied.

Format:

```
re
```

Table 5.13 RESET Command Parameter

Parameter	Type	Description
None		

Example:

To reset the H8/xxxx:

```
>re (RET)
```

5.2.13 STATUS: STS

Description:

Displays emulator status information.

Format:

```
sts
```

Table 5.14 STATUS Command Parameter

Parameter	Type	Description
None		

Example:

To display status information of the emulator:

```
>sts (RET)
```

The display format is as follows:

```
>sts
Emulator Status
Connected to:         H8/xxxx E10T Emulator (E10T PC Card Driver)
CPU                   H8/xxxx
Run status            Break
Cause of last break   BREAK KEY
Run time count        0H:0M:0S:0MS
Emulator mode         Normal
Big endian
```

Note: The items displayed with this command vary with the product. For the display specifications of each product, refer to the on-line help.

5.2.14 STEP_INTERRUPT: SI

Description:

Sets or displays the enable or disable status of interrupts during step execution. When interrupts occur when they are disabled, execution stops at the top address of the interrupt routine.

Format:

```
si
```

Displays the enable or disable status of interrupts during step execution.

```
si <mode>
```

Sets the enable or disable status of interrupts during step execution.

Table 5.15 STEP_INTERRUPT Command Parameter

Parameter	Type	Description
<mode>	Keyword	Enables or disables interrupts during step execution. Set either of the following: **enable**: Enables interrupts. **disable**: Disables interrupts.

Example:

To enable interrupts during step execution:

```
>si enable (RET)
```

To display interrupt status during step execution:

```
>si (RET)
Emulator step interrupt mode = ENABLE
```

5.2.15 TRACE_DISPLAY: TD

Description:

Displays acquired trace information.

Format:

```
td
```

Table 5.16 TRACE_DISPLAY Command Parameter

Parameter	Type	Description
None		

Example:

To display acquired trace information:

```
>td (RET)
```

The display format is as follows:

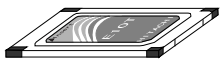
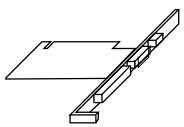
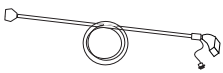
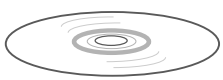
```
>td
   IP          TYPE            ADDR        MNEMONIC      OPERAND
-D'xxxx        BRANCH        00001010        JSR           @ER0
-D'xxxx        BRANCH        00001200        JMP           @ER1
(a)            (b)           (c)             (d)           (e)
```

(a) Instruction pointer (signed decimal)
(b) Type of branch source
BRANCH: Branch source
(c) Address of instruction word
(d) Instruction mnemonic
(e) Instruction operand

6.1 Components of the Emulator

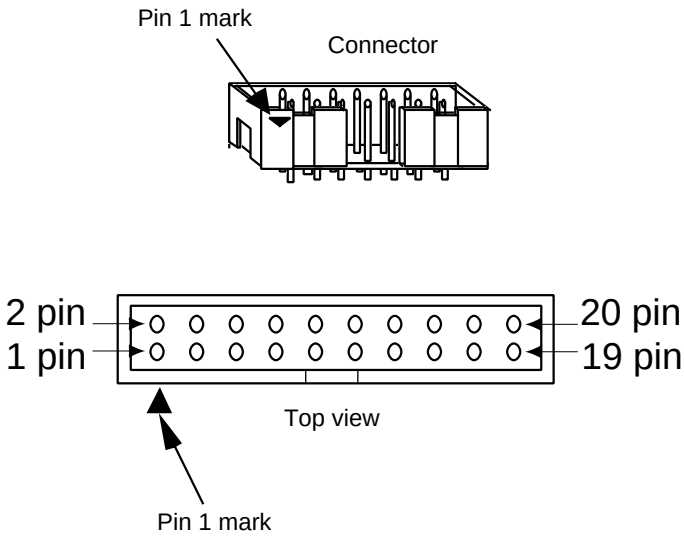
The H8/3048F-ONE E10T emulator supports the H8/3048F-ONE (F-ZTAT only). Table 6.1 lists the components of the emulator.

Table 6.1 Components of the Emulator (HS3048BTCM01H or HS3048BTCI01H)

Classi-fication	Component	Appearance	Quan-tity	Remarks
Hard-ware	Card emulator HS3048BTCM02H (Model: HS0005TCM01H), HS3048BTCI02H (Model: HS0005TCI01H)	 or 	1	HS3048BTCM01H (PCMCIA: 20-pin type): Depth: 85.6 mm, Width: 54.0 mm, Height: 5.0 mm, Mass: 28.0 g HS3048BTCI01H (PCI: 20-pin type): Depth: 122.0 mm, Width: 96.0 mm, Mass: 78.0 g
	User system interface cable		1	HS3048BTCM01H (PCMCIA: 20-pin type): Length: 80.0 cm, Mass: 46.0 g HS3048BTCI01H (PCI: 20-pin type): Length: 150.0 cm, Mass: 90.0 g
Soft-ware	H8/3048F-ONE E10T emulator setup program, H8/3048F-ONE E10T Emulator User's Manual, and Hitachi Debugging Interface User's Manual		1	HS3048BTCM01SR, HS3048BTCM01HJ, HS3048BTCM01HE, HS6400DIIW5SJ, and HS6400DIIW5SE (provided on a CD-R)

6.2 Pin Arrangement of the Connector

Figure 6.1 shows the pin arrangement of the connector.



Pin No.	Signal
1	#RES
2	Vss
3	FWE
4	Vss
5	MDO
6	Vss
7	MD1
8	Vss
9	MD2
10	Vss
11	N.C.
12	Vss
13	N.C.
14	Vss
15	P91/TxD1
16	Vss
17	P93/RxD1
18	Vcc
19	P95/SCK1
20	Vcc

Notes: 1. The symbol (#) means that the signal is active-low.
2. Vss (pin 2) detects connection or disconnection with the E10T user system.
3. Pin 20 is Vcc; however, it can be N.C. if an on-board flash writer (including the third party's product) is not used.

Figure 6.1 Pin Arrangement of the Connector

Figure 6.2 shows connection of the emulator.

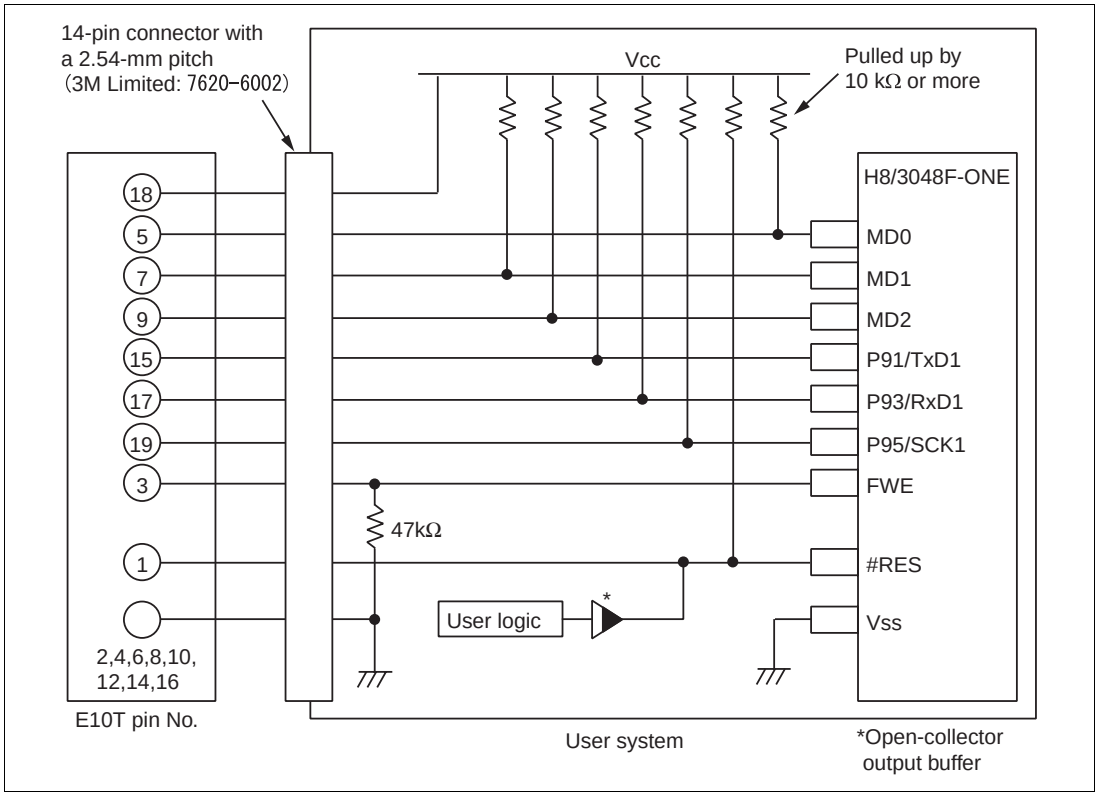


Figure 6.2 Connection of the Emulator (Mode 7)

Notes: 1. P91, P93, and P95 are used by the emulator. Pull up and connect the emulator and MCU pins.

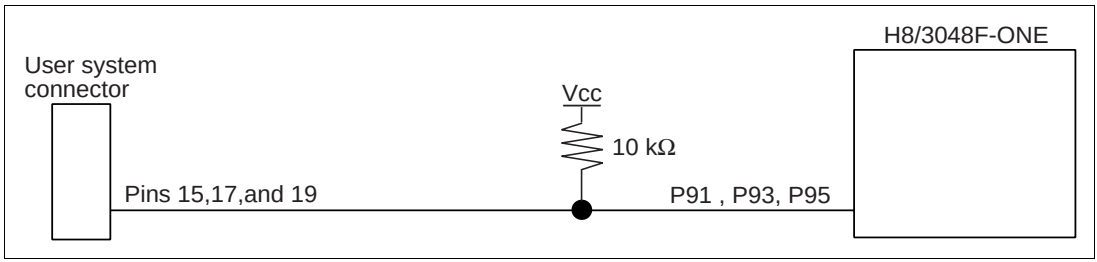


Figure 6.3 Connection of Emulator and P91, P93, and P95 Pin

2. **The FWE signal is used for forced break control by the emulator. Connect the emulator and MCU pins directly.**

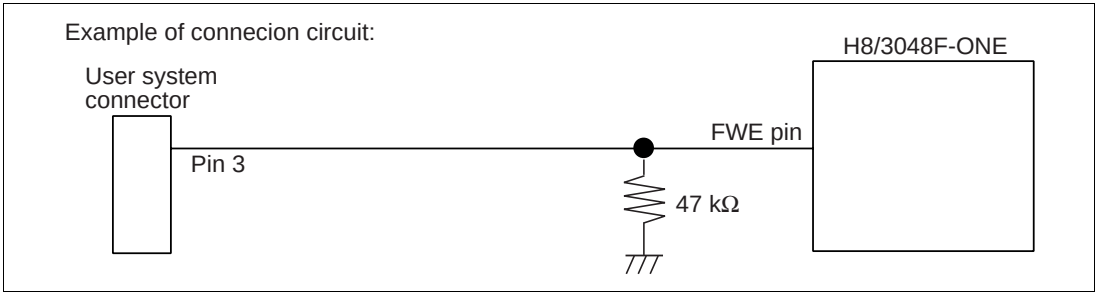


Figure 6.4 Direct Connection of Emulator and FWE Pin

3. **The #RES pin is used by the emulator. Create the following circuit by connecting the reset input from the emulator with an open-collector output buffer to accept the reset control on the user system:**

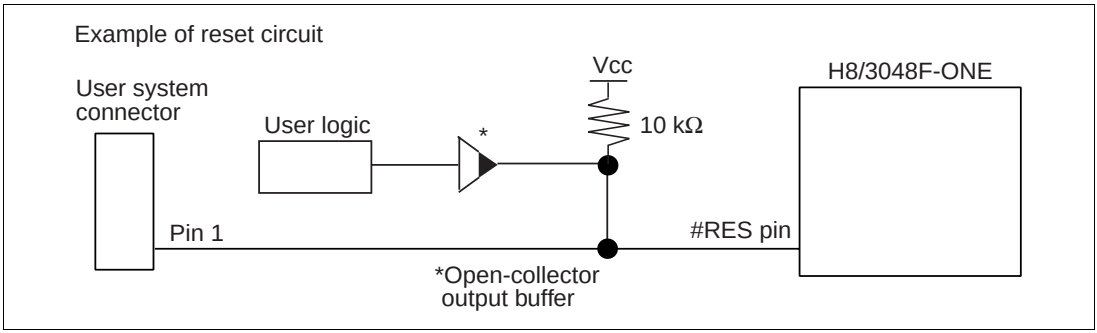


Figure 6.5 Example of Reset Circuits

4. **MD0 to MD2 pins are used by the emulator at power-on reset. Connect MD0 to MD2 pins as shown in figure 6.6.**

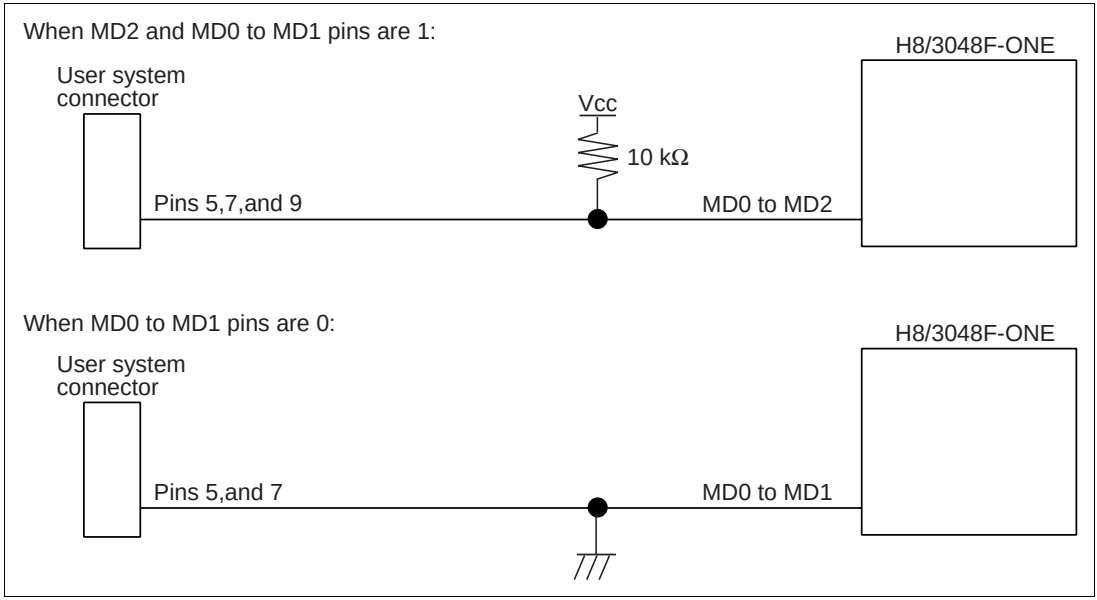


Figure 6.6 Example of Circuits for Operating Mode Setting Pins

5. **Connect Vss and Vcc with the Vss and Vcc of the MCU, respectively.**
6. **Connect nothing with N.C.**
7. **The amount of voltage permitted to input to Vcc must be within the guaranteed range of the microcomputer. The H8/3048F-ONE has 5.5- to 4.5-V and 3.0- to 3.6-V input voltages. Use the correct input voltage depending on the microcomputer.**

8. **Figure 6.7 shows the interface circuit in the emulator. Use this figure as a reference when determining the pull-up resistance value.**

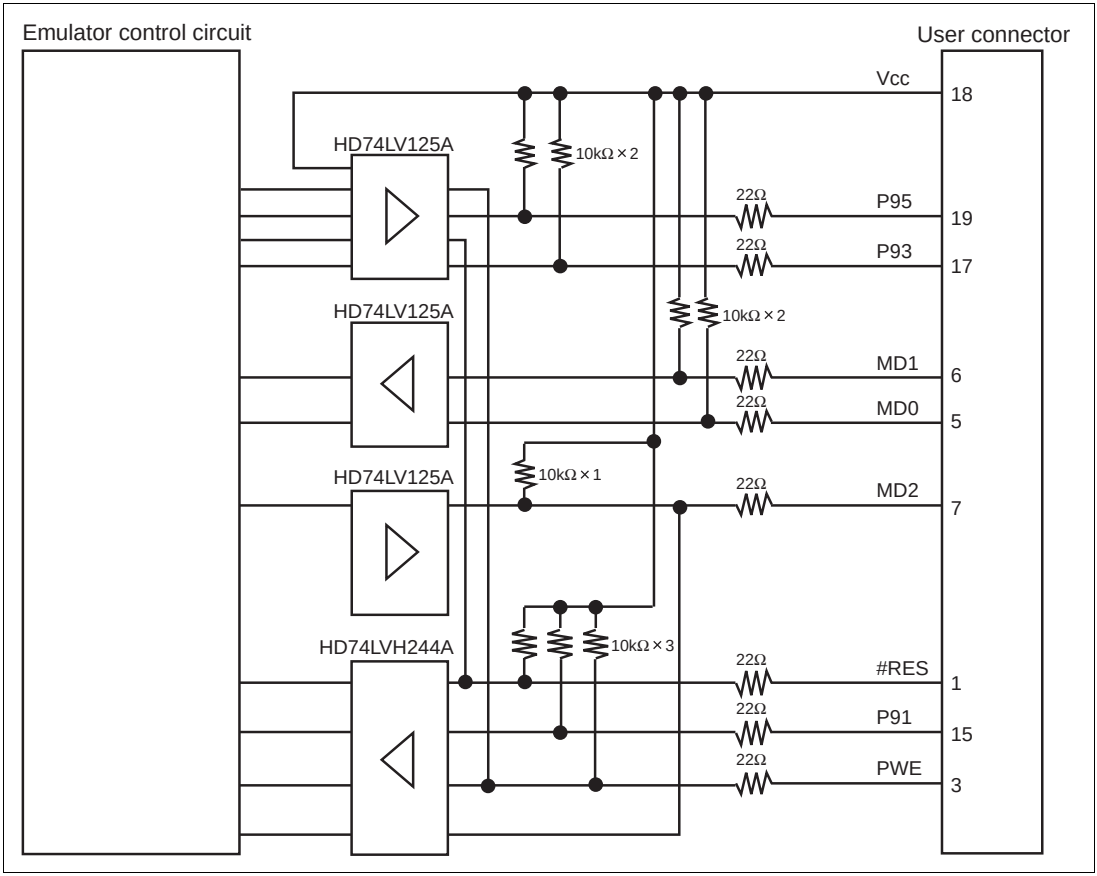


Figure 6.7 Interface Circuit in the Emulator (Reference)

6.3 Differences between the H8/3048F-ONE and the Emulator

1. Operating Mode

   The emulator supports the following operating modes:

   - Mode 5
   - Mode 6
   - Mode 7

2. When the emulator system is initiated, it initializes the general registers and part of the control registers as shown in table 6.2.

Table 6.2 Register Initial Values at Emulator Power-On

Register	When Using H8/3048F-ONE
PC	Reset vector value in the vector address table
ER0 to ER6	Undefined
ER7 (SP)	H'FFFF10 (for mode 6) H'FFF10 (for mode 5 or mode 7)
CCR	1 for I mask, and others undefined

3. Low-Power Mode

   During a user program break, the CPU operating frequency is forced to ϕ for high-speed operation.

4. RES signal

   The H8/3048F-ONE RES signal is only valid during user program execution started with clicking the GO or STEP-type button. If this signal is input from the user system in command input wait state, it is not sent to the H8/3048F-ONE.

Note: Do not start user program execution or access the memory while control input signal (RES) is being low. A TIMEOUT error will occur. The error will also occur when (RES) is low during a break.

5. System Control Register

   In the emulator, the internal I/O registers can be accessed from the [I/O registers] window. However, be careful when accessing the system control register. The emulator saves the register value of the system control register at a break and returns the value when the user program is executed. Since this is done during a break, do not rewrite the system control register in the [I/O Registers] window.

6. Memory Access during Emulation

   If the memory contents are referenced or modified during emulation, realtime emulation cannot be performed because the user program is temporarily halted.

7. The emulator communicates with the H8/3048F-ONE by using the P95/SCK1, P93/RxD1, P91/TxD1, RES, FWE, and M2 to M0 pins. These pins cannot be used, except for RES and M2 to M0 pins.

8. The power consumed by the MCU can reach several mA. This is because the user power supply drives one HD74LV125A to make the communication signal level match the user-system power-supply voltage. The power consumed rises little during user program execution since the emulator does not perform communication; it rises more during a break.

9. The emulator uses a two-word stack pointer for values stored on a user program break. Therefore, the stack area must accept two-word addresses.

6.4 The H8/3048F-ONE E10T Emulator Functions

Notes: 1. Do not use an MCU that has been used for debugging.
2. If the flash memory is rewritten many times, and the emulator is left for a few days, data may be lost due to retention problems.
3. If the flash memory is rewritten many times, the data will not be erased. If an error message is displayed, exchange the MCU for a new one.

6.4.1 Emulator Driver Selection

Table 6.3 shows drivers which can be selected in the [E10T Driver Details] dialog box.

Table 6.3 Type Name and Driver

Type Name	Driver
HS3048BTCM01H	E10T PC Card Driver 2
HS3048BTCI01H	E10T PCI Card Driver 2

6.4.2 Hardware Break Functions

Hardware Break Conditions: In the H8/3048F-ONE E10T emulator, two break conditions can be set (Break Condition 1,2). Table 6.4 lists the items that can be specified.

Table 6.4 Hardware Break Condition Specification Items

Items	Description
Address bus condition	Breaks when the MCU address bus value matches the specified value.
Data bus condition	Breaks when the MCU data bus value matches the specified value. Bit or word can be specified as the access data size.
Read or write condition	Breaks in the read or write cycle.

Table 6.5 lists the combinations of conditions that can be set in the [Break condition 1] and [Break condition 2] dialog boxes.

Table 6.5 Conditions Set in [Break condition 1] and [Break condition 2] Dialog Boxes

	Condition		
Dialog Box	Address Bus Condition	Data Condition	Read or write condition
[Break condition 1]	O	O	O
[Break condition 2]	O	x	x

Note: O: Can be set by checking the radio button in the dialog box.
X: Cannot be set in the dialog box.

Table 6.6 lists the combinations of conditions that can be set by the BREAKCONDITION_SET command.

Table 6.6 Conditions Set by BREAKCONDITION_SET Command

	Condition		
Channel	Address Bus Condition (<addropt> option)	Data Condition (<dataopt> option)	Read or write condition (<r/wopt> option)
Break condition 1	O	O	O
Break condition 2	O	x	x

Note: O: Can be set by the BREAKCONDITION_SET command.
X: Cannot be set by the BREAKCONDITION_SET command.

Notes on Setting the Break Condition:

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of Break Condition 1 are disabled.
2. Setting of Break Condition 1 is disabled when an instruction to which a software breakpoint has been set is executed.
3. When step over function is used, the settings of software breakpoint and Break Condition 1 are disabled.

6.4.3 Notes on Setting the [Breakpoint] Dialog Box

1. When an odd address is set, the address is rounded down to an even address.
2. A software break is accomplished by replacing instructions. Accordingly, it can be set only to the flash memory or RAM area. However, a software break cannot be set to the following addresses:
   - An area other than flash memory or RAM
   - An instruction in which Break Condition 1 is satisfied
3. During step execution, a software breakpoint is disabled.
4. A condition set at Break Condition 1 is disabled immediately after starting execution when an instruction at a software breakpoint is executed. A break does not occur even if a condition of Break Condition 1 is satisfied immediately after starting the execution.
5. When execution resumes from the breakpoint address after the program execution stops at the software breakpoint, single-step execution is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.
6. Settings of software breakpoint and Break Condition 1 are invalid while the STEP OVER function is being used.

6.4.4 Trace Function

The trace function in the H8/3048F-ONE E10T emulator uses the branch-instruction trace function in the MCU. It displays the branch-source address or the mnemonic, and operand can be acquired in realtime.

6.4.5 Notes on HDI

1. Note on Moving Source File Position after Creating Load Module

   When the source file is moved after creating the load module, the [Open] dialog box may be displayed to specify the source file during the debugging of the created load module. Select the corresponding source file and click the [Open] button.

2. Source-Level Execution

   — Source file

   Do not display source files that do not correspond to the load module in the program window. For a file having the same name as the source file that corresponds to the load module, only its addresses are displayed in the program window. The file cannot be operated in the program window.

   — Step

   Even standard C libraries are executed. To return to a higher-level function, enter Step Out. In a for statement or a while statement, executing a single step does not move execution to the next line. To move to the next line, execute two steps.

3. Operation During Accessing Files

   Do not perform other operations during saving in the [Load Program], [Verify Memory], [Save Memory], or [Trace] window because this will not allow correct saving to be performed.

4. Source Window at Program Change

   When a program being displayed in the source window is changed and the source file and load module are reloaded, close and reopen the source window once. If the window is not closed and reopened, the display will be illegal.

5. Watch

   — Local variables at optimization

   Depending on the generated object code, local variables in a C source file that is compiled with the optimization option enabled will not be displayed correctly. Check the generated object code by displaying the [Disassembly] window.
   If the allocation area of the specified local variable does not exist, displays as follows.

   Example: The variable name is asc.

   asc = ? - target error 2010 (xxxx)

   — Variable name specification

   When a name other than a variable name, such as a symbol name or function name, is specified, no data is displayed.

Example: The function name is main.

main =

— Array display

When array elements exceed 1000, elements from after 1000 will not be displayed.

6. Memory Load Function

When [Load...] is selected from the [Memory] menu, the Memory Load function can be used although it takes time to download. It is recommended that the File Load function ([Load Program...] selected from the [File] menu) is used to load the S-type file.

Note: The File Load function deletes the debugging information of the previously loaded program. When other load modules are loaded after the program to be debugged has been loaded, use the following sequence: Link the program to be debugged with the Sdebug option and save the debugging information in another file. Load the debugging information file after all the load modules have been loaded.

7. Line Assembly

— Input radix

Regardless of the Radix setting, the default for line assembly input is decimal. Specify H' or 0x as the radix for a hexadecimal input.

8. Command Line Interface

— Batch file

To display the message "Not currently available" while executing a batch file, enter the sleep command. Adjust the sleep time length which differs according to the operating environment.

Example: To display "Not currently available" during memory_fill execution:

sleep d'3000

memory_fill 0 ffff 0

— Overwrite file

In Command Line Interface, a file having the same name as the output file is overwritten without asking the user.

— File specification by commands

The current directory may be altered by file specifications in commands. Absolute paths should be used to specify the files in a command file so that the current directory alteration is not affected.

Example: FILE_LOAD C:\\HEW\\HDI5\\E10T\\3048fone
\\Tutorial\\TUTORIAL.ABS

9. Note on Initiating HDI

   When the emulator is initiated by using another card emulator after it has been initiated by using the PCI card emulator, delete the [TARGET] line from the C:\windows\HDI.INI file.

10. Usage with Another HDI

— Automatic load of session files

Since the emulator cannot use another HDI, re-install this HDI whenever another HDI has been previously installed is used.
If another HDI has been used, initiate this HDI with “Run” as follows without using the session files.

```
<Directory path name in which HDI is installed>\hdi /n (RET)
```

/n initiates the HDI without loading the recently used session files.

If there is another session file in the different debug platform, the following error message is displayed:

```
invalid target system: <recently used debug platform name>
```

11. [Select Function] dialog box

    This HDI does not support software breakpoint setting in the [Select Function] dialog box (described in section 10, Selecting Functions, in the Hitachi Debugging Interface User's Manual).

12. Memory Save During User Program Execution

    Do not execute memory save verifying during user program execution.

13. Loading Motorola S-type files

    This HDI does not support Motorola S-type files with only the CR code (H'0D) at the end of each record. Load Motorola S-type files with the CR and LF codes (H'0D0A) at the end of each record.

14. [Memory] window

    If the following memory contents are displayed, they will be incorrect.

    Word access from address 2n + 1

    Longword access from address 4n + 1, 4n + 2, or 4n + 3

15. Scrolling Window During User Program Execution

    Do not scroll the [Memory] and [Disassembly] windows by dragging the scroll box during user program execution. This generates many memory reads causing the user program to stop execution until the memory reads have been completed.

16. [I/O Registers] window

— Display and modification

- Do not change values in the [I/O Registers] window because the emulator uses the address break controller.
- Note that the E10T emulator does not support the display of the invalid module or bit information in the [I/O Registers] window (described in section 8, Displaying Variables, in the Hitachi Debugging Interface User's Manual).
- Verify
  In the [I/O Registers] window, the verify function of the input value is invalid.

17. Note on [Registers] Window Operation During Program Execution

Although a dialog box is displayed in which the register contents can be changed by double-clicking the [Registers] window, do not change the register contents during program execution.

18. Note on Radix in the [Register] Dialog Box

The default input radix in the [Register] dialog box is hexadecimal irrespective of the Radix display. When a radix other than a hexadecimal is input, specify the prefix code such as B'. After the value has been input in the [Register] dialog box, the Radix setting is changed to hexadecimal. When the radix other than a hexadecimal is used as a default, reset the Radix display.

19. Software Break

— Session file

When the software breakpoint address set in the session file is H'0, the breakpoint will not be set. If the address set as the breakpoint is wrong, the error message is not output. The breakpoint is registered as DISABLE in the [Breakpoints] window.

— [Breakpoints] window

During user program execution, it is impossible to jump from the breakpoint to the source or address line in the [Source] or [Disassembly] window by using [Go to Source] in the popup menu displayed in the [Breakpoints] window.

— When the software breakpoint is set in the flash memory area, the program is written to the flash memory each time the user program is executed. At this time, note that the number of rewritable times will be decreased.

— Breakpoint cancellation

When the contents of the software breakpoint address is modified during user program execution, the following message is displayed when the user program stops.

BREAKPOINT IS DELETED A=xxxxxxxx

If the above message is displayed, cancel all software breakpoint settings with the [Delete All] or [Disable] button in the [Breakpoints] window.

20. Number of Software Breakpoint and [Stop At] Settings in the [Run...] Menu

    The maximum number of software breakpoints and [Stop At] settings allowed in the [Run...] menu is 255. Therefore, when 255 software breakpoints are set, specification by [Stop At] in the [Run...] menu becomes invalid. Use the software breakpoints and [Stop At] in the [Run...] menu with 255 or less total settings.

21. Note on RUN-TIME Display

    The execution time of the user program displayed in the [Status] window is not a correct value since the timer in the host computer has been used.

22. Note on Displaying COMMUNICATION TIMEOUT ERROR

    If COMMUNICATION TIMEOUT ERROR is displayed, the emulator cannot communicate with the MCU. Select [Initialize] from the [File] menu to initialize the emulator.

23. Note on Downloading Program

    In the [Load Program] dialog box, which is opened when [Load Program…] is selected, the verify function is invalid. After downloading the program, perform verify in the [Verify S-Record File with Memory] dialog box, which is opened when [Verify] is selected from the [Memory] menu.

24. Support of Double Float Format

    In the following memory operations, the double float format is not supported:

    — [Fill Memory] dialog box
    — [Search Memory] dialog box
    — MEMORY_FILL command

      The [Format] specification in the [Copy Memory] dialog box is ignored. Memory is copied in a byte unit.
    — Double float display at little endian operation

25. Note on Continuous Step Execution

    When the step is continuously executed by selecting [Step...] from the [Run] menu, do not use the software breakpoint because this will cause the HDI to abnormally operate.

26. Note on Using the [Run program] Dialog Box

    When [Run...] is selected from the [Run] menu to specify the stop address, there is the following note:

    — When the breakpoint that has been set as Disable is specified as the stop address, note that the breakpoint becomes Enable when the user program stops.